ȘT. DUMITRESCU
V. MARINOIU

I. DUMITRESCU
M. FLOROVICI

APLICAȚII INGINEREȘTI ALE CALCULATOARELOR

CALCULATOARE DE PROCES

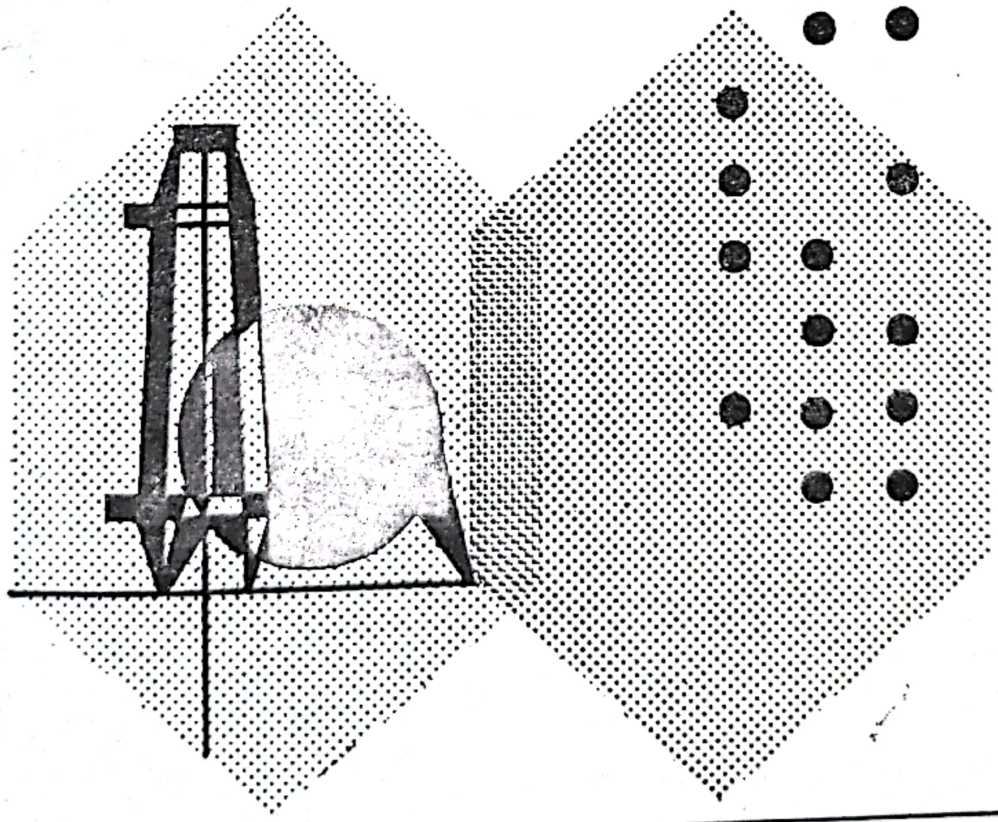
Scanned with OKEN Scanner

MINISTERUL EDUCAȚIEI ȘI ÎNVĂȚĂMÎNTULUI

Prof. dr. ing. STELIAN DUMITRESCU
Prof. dr. ing. ION I. DUMITRESCU
Conf. dr. ing. VASILE MARINOIU
Ing. MIHAI FLOROVICI

APLICAȚII INGINEREȘTI ALE CALCULATOARELOR

CALCULATOARE DE PROCES



EDITURA DIDACTICĂ ȘI PEDAGOGICĂ, BUCUREȘTI

Scanned with OKEN Scanner

Scanned with OKEN Scanner

Cuprinsul volumului 1

APLICAȚII INGINEREȘTI ALE CALCULATOARELOR CU EXEMPLE DIN INDUSTRIA PETROLIERĂ ȘI PETROCHIMICĂ

Capitolul 1. PRELUCRAREA AUTOMATĂ A DATELOR (43 pagini)

1.1. Configurația sistemelor de calcul automat
1.2. Reprezentarea informației in echipamentele automate de calcul
1.3. Principiile prelucrării automate a datelor
1.4. Limbaje algoritmice de programare
1.5. Sisteme de operare

Capitolul 2. BAZELE PROGRAMĂRII ÎN FORTRAN (82 pagini)

2.1. Structura programelor in FORTRAN
2.2. Elementele de bază ale limbajului FORTRAN
2.3. Instrucțiuni de atribuire
2.4. Instrucțiuni de salt
2.5. Instrucțiuni de ciclare
2.6. Instrucțiuni de intrare/ieșire
2.7. Instrucțiuni proprii funcțiilor și subprogramelor
2.8. Directive de comandă
2.9. Fișiere utilizabile în FORTRAN

Capitolul 3. INIȚIERE ÎN PL/1 (49 pagini)

3.1. Elementele unui program PL/1
3.2. Date
3.3. Expresii
3.4. Instrucțiuni
3.5. Instrucțiuni de intrare/ieșire pentru fișierele standard
3.6. Blocuri și activarea blocurilor
3.7. Alocarea memoriei
3.8. Recunoașterea numelor
3.9. Subrutine și funcții
3.10. Atributul PICTURE
3.11. Instrucțiunea ON
3.12. Fișiere pe discuri și benzi magnetice
3.13. Exemplu de program PL/1

Capitolul 4. METODE NUMERICE (216 pagini)

4.1. Bazele metodelor numerice
4.2. Prelucrarea datelor experimentale
4.3. Rezolvarea numerică a ecuațiilor neliniare

28—27

4.4. Rezolvarea numerică a sistemelor de ecuaţii algebrice liniare
4.5. Rezolvarea numerică a ecuaţiilor diferenţiale ordinare
4.6. Noţiuni cu privire la rezolvarea numerică a problemelor cu condiţii la limită în care intervin ecuaţii cu derivate parţiale

Capitolul 5. APLICAŢII (40 pagini)
5.1. Studiul proceselor de filtraţie staţionare din zăcămintele petrolifere
5.2. Sinteza profilului organelor de obturare ale robinetelor de reglare
5.3. Determinarea eforturilor din elementele turlelor de foraj de tip „A"
5.4. Calculul compoziţiei chimice a produselor rezultate din reacţia de piroliză a etanului

Cuprinsul volumului 2

APLICAŢII INGINEREŞTI ALE CALCULATOARELOR — OPTIMIZĂRI

Capitolul 6. APLICAŢIILE PROGRAMĂRII LINIARE (94 pagini)
6.1. Modelarea prin programe liniare de optimizare
6.2. Metode de rezolvare a programelor liniare
6.3. Pachete de programe specializate pentru rezolvarea problemelor de programare liniară

Capitolul 7. APLICAŢIILE PROGRAMĂRII NELINIARE (40 pagini)
7.1. Programe neliniare
7.2. Algoritmi pentru programele neliniare
7.3. Un exemplu de aplicaţie — optimizarea procesului de piroliză a etanului

Capitolul 8. APLICAŢIILE PROGRAMĂRII DINAMICE (72 pagini)
8.1. Introducere în programarea dinamică
8.2. Exemple de aplicaţii
8.3. Principiile modelării dinamice
8.4. Modelarea dinamică prin metoda lui Forrester
8.5. Un exemplu de aplicaţie al metodei Forrester
8.6. Algoritm de conducere optimală

Capitolul 9. OPTIMIZĂRI PRIN METODE STATISTICE ALE CERCETĂRII OPERAŢIONALE (74 pagini)
9.1. Aplicaţiile firelor de aşteptare
9.2. Aplicaţiile teoriei stocurilor
9.3. Aplicaţiile teoriei echipamentului

Scanned with OKEN Scanner

PREFAȚĂ

Cel de al treilea volum (și ultimul) al lucrării **Aplicații inginerești ale calculatoarelor** *se referă la un domeniu de mare eficiență al utilizării tehnicii de calcul și anume acela al folosirii calculatoarelor automate la conducerea în timp real a proceselor industriale, adică la* aplicațiile calculatoarelor de proces.

Folosirea calculatoarelor la supravegherea continuă a proceselor și la conducerea instalațiilor tehnologice a permis realizarea unui înalt nivel de automatizare în întreprinderile industriale, cu performanțe deosebite de productivitate și calitate a producției, cu consumuri specifice minime de materii, materiale și energie. Utilizarea calculatoarelor de proces permite implementarea unor sisteme de automatizare complexă chiar și în cazul unor procese foarte rapide și cu un mare număr de parametri, procese greu de controlat cu mijloacele convenționale (adică fără sisteme de calcul).

După cum se știe, automatizarea proceselor de producție și introducerea pe scară tot mai largă a tehnicii de calcul în conducerea activității industriale sînt factori determinanți ai progresului tehnic și constituie direcții principale ale revoluției tehnico-științifice contemporane. Ca urmare, și în țara noastră, în ultimii zece ani, s-au întreprins numeroase acțiuni pentru promovarea aplicațiilor calculatoarelor de proces. În special în actualul cincinal — apreciat de tovarășul Nicolae Ceaușescu, secretarul general al partidului, ca fiind cincinalul afirmării ample a revoluției tehnico-științifice în toate sectoarele de activitate — sînt prevăzute măsuri concrete de extindere a automatizării prin utilizarea eficientă a calculatoarelor de proces. Astfel, în anii 1976—1980 vor fi realizate sisteme tipizate de conducere a proceselor tehnologice (continue și discontinue) cu ajutorul calculatoarelor automate, precum și sisteme de pachete de programe tipizate pentru procesele industriale, în special în industria chimică, petrochimică, energetică, metalurgică, a materialelor de construcții și a construcțiilor de mașini. Se va extinde gama acestor sisteme cu o serie de variante tipizate de echipamente pentru supravegherea centralizată și conducerea proceselor tehnologice cu ajutorul unor noi tipuri de minicalculatoare. Un început s-a și făcut prin realizarea în anul 1975 a primului calculator românesc de proces: „FELIX" C—32 P.

În acest context, în țara noastră există un larg interes pentru această relativ nouă aplicație a tehnicii de calcul, mai ales în rîndul inginerilor tehnologi și de exploatare, al cadrelor de conducere și al economiștilor din industrie, care sînt confruntați din ce în ce mai des nu numai cu noile noțiuni din acest domeniu, ci și cu proiecte și aplicații efective ale calculatoarelor de proces. Se face simțită, astfel, nevoia unei lucrări care să prezinte noțiunile de bază,

Scanned with OKEN Scanner

echipamentele și aplicațiile din acest nou domeniu al tehnicii, familiarizînd pe cel ce nu este specialist în sistemele de calcul cu problemele legate de conducerea cu calculatoarele de proces, pe care le are conectate la instalațiile tehnologice din întreprinderea (secțiile) unde lucrează.

Răspunzînd acestei necesități, de informare a unui cerc larg de ingineri tehnologi (petroliști, petrochimiști, chimiști, energeticieni, metalurgiști, constructori de mașini etc.) și economiști care lucrează în industrie, cartea este destinată în primul rînd studenților din institutele tehnice și în special studenților de la Institutul de petrol și gaze. De aceea, ea este concepută ca un manual didactic menit să asigure materialul de studiu la cursurile în a căror programă sînt incluse capitole referitoare la automatizarea complexă și la aplicațiile calculatoarelor de proces.

Avînd în vedere acest scop, în cele trei capitole ale prezentului volum sînt expuse următoarele părți fundamentale ale teoriei și practicii utilizării calculatoarelor de proces : principiile folosirii sistemelor de calcul la conducerea proceselor industriale, configurația sistemelor de echipamente (cu exemple de calculatoare de proces utilizate în țara noastră), sistemele de programe și de operare specifice calculatoarelor de proces, precum și cîteva din cele mai reprezentative aplicații ale conducerii proceselor industriale cu ajutorul calculatoarelor automate în diferite domenii (exploatarea zăcămintelor de hidrocarburi fluide, metalurgie, construcții de mașini și petrochimie).

Conținutul și modul de expunere este subordonat intenției noastre de a întocmi o lucrare unitară, menită să inițieze în domeniul utilizării calculatoarelor de proces pe cel ce nu este specialist (prin formație) în tehnica de calcul, în așa fel încît să înțeleagă particularitățile exploatării instalațiilor tehnologice conduse cu ajutorul calculatoarelor. Sperăm ca în acest mod să putem mări interesul față de acest domeniu de vîrf al automaticii/informaticii și să lărgim sfera aplicațiilor practice ale calculatoarelor de proces în țara noastră.

Încheind, cu acest al III-lea volum, lucrarea „Aplicații inginerești ale calculatoarelor", nu putem să nu dăm curs sentimentelor noastre de respect față de tovarășul prof. dr. doc. Edmond Nicolau, care în calitatea sa de referent al tuturor celor trei volume, ne-a ajutat efectiv prin exigența și sfaturile sale, la realizarea acestei lucrări.

Mulțumim călduros Editurii didactice și pedagogice și Întreprinderii poligrafice din Cluj-Napoca care s-au străduit ca, din manuscrise depuse la intervale de timp relativ mari (de-a lungul a trei ani), să redacteze și să tipărească într-o formă unitară cele trei volume ale acestei lucrări.

Prof. dr. ing. STELIAN DUMITRESCU, prof. dr. ing. I. DUMITRESCU

Ploiești,
septembrie 1977.

Scanned with OKEN Scanner

CUPRINS

Prefață 5

Capitolul 10. UTILIZAREA SISTEMELOR DE CALCUL LA CONDUCEREA PROCESELOR INDUSTRIALE 11

10.1. Principii fundamentale 11

Noțiuni generale 12

Principiul acțiunii după abatere 14

Principiul acțiunii după perturbație 15

Principiul acțiunii combinate 15

Structura generală a sistemelor de conducere 16

10.2. Metode și mijloace de conducere a proceselor tehnologice 17

Conducerea cu mijloace tradiționale 17

— *Conducerea directă de către om* 17

— *Conducerea cu ajutorul regulatoarelor* 18

— *Conducerea cu ajutorul instalațiilor de control și reglare centralizate* 18

Conducerea cu ajutorul calculatoarelor de proces 20

— *Conducerea cu ajutorul mașinilor de control și reglare centralizate* 21

— *Conducerea cu calculatoare în regim de consultant* 22

— *Conducerea cu calculatoare conectate la proces* 24

Particularitățile funcționale și structurale ale calculatoarelor de proces . . 26

— *Cerințe și particularități funcționale* 26

— *Structura generală a calculatoarelor de proces* 27

Echipamentul de interfață proces-calculator 30

— *Canale de intrare/ieșire* 30

— *Canale de intrare analogică* 30

— *Canale de intrare numerice și logice* 32

— *Canale de ieșire* 34

— *Canale și echipamente terminale* 37

Alegerea componentelor calculatoarelor de proces 39

10.3. Introducere în practica modelării și identificării proceselor tehnologice . 41

Problemele generale ale modelării 41

Elementele esențiale ale unui model 42

Variabilele și restricțiile modelelor 43

— *Variabile independente* 43

— *Variabile dependente* 44

— *Variabile de stare* 44

— *Variabila sau funcția de performanță* 45

— *Restricțiile proceselor și modelelor* 47

Tipuri de modele și premise de elaborare 48

Scanned with OKEN Scanner

— *Modele funcțional-procedurale* 49
— *Modele matematice* 51
Premisele modelării deterministe 52
— *Ecuația de conservare a masei* 53
— *Ecuațiile de conservare a masei pe componenți* 54
— *Ecuația de conservare a energiei* 56
— *Ecuațiile de conservare a stării de mișcare* 58
— *Ecuațiile de transport* 59
Prelucrarea și perfecționarea modelelor 62
Noțiuni referitoare la identificarea proceselor 63
Metode de estimare a modelelor 66
— *Metoda regresiei* 66
— *Metoda de corelație* 69
— *Metode iterative de gradient* 72
10.4. Funcțiunile calculatoarelor de proces 76
Supravegherea proceselor și echipamentelor 77
— *Supravegherea aparatelor de măsurat și analizoarelor* 77
— *Supravegherea stării echipamentelor tehnologice* 78
— *Calculul unor indici sintetici* 80
— *Înregistrarea operativă a activităților și evenimentelor* 81
— *Semnalizarea deficiențelor și protecția instalațiilor* 83
Conducerea automată a proceselor 85
— *Conducerea secvențială a operațiilor* 85
— *Comanda și reglarea automată multiplă* 88
Optimizarea automată a activităților tehnice și economice 92
— *Noțiuni generale* 92
— *Optimizarea staționară fără restricții* 94
— *Optimizarea staționară cu restricții* 99
Optimizarea staționar-dinamică evolutivă 102
— *Strategii de gradient* 104
Optimizarea adaptivă 107
— *Adaptarea la schimbarea parametrilor procesului* 108
— *Adaptarea la restricții* 110
Optimizarea dinamică 111
— *Principiul maximului* 112
— *Programarea dinamică* 114
10.5. Noțiuni cu privire la proiectarea și implementarea sistemelor de conducere cu calculatoare 116
Obiective și etape principale 116
Implicațiile implementării unui *SCC* 120
Capitolul 11. SISTEME DE ECHIPAMENTE 123
11.1. Concepția generală a calculatoarelor de proces 123
Configurația calculatoarelor de proces 124
— *Configurația generală* 125
— *Exemple* 127
Caracteristicile calculatoarelor de proces 137
— *Cerințele funcționale ale unui sistem de proces* 138
— *Condițiile funcționale impuse calculatoarelor de proces* 140
11.2. Prelevarea datelor 147
Senzori — extragerea datelor 148

Scanned with OKEN Scanner

— *Senzori analogici* 149
— *Senzori numerici (digitali)* 157
Transmisia semnalelor 164
— *Perturbații* 165
— *Transmisia semnalelor analogice* 168
— *Transmisia semnalelor numerice* 170
Executori — introducerea comenzilor 170
— *Executori analogici* 170
— *Executori numerici (digitali)* 171
11.3. Sistemul de intrare/ieșire 174
Tipuri de schimburi de informație între unitatea centrală de prelucrare și exterior 176
— *Realizarea legăturilor* 176
— *Liniile de intrare/ieșire programată* 176
— *Linia pentru întreruperi prioritare* 177
— *Linia de acces direct la memoria internă* 179
Interfața industrială 179
— *Generalități* 180
— *Structura interfeței industriale* 182
— *Exemplu: cuplorul de proces al sistemului FELIX C—32P* . . . 185
— *Exemplu: interfața standard IBM 1827* 196
11.4. Aplicații 202
Exemplu: tehnica analizelor 203
Exemplu: analiza comportării în exploatare a zăcămintelor petroliere. . . 205
Exemplu: sistem de conducere pentru liniile de prelucrări mecanice . . 209
Capitolul 12. SISTEME DE PROGRAME 211
12.1. Sisteme de operare 212
Definiția sistemului de operare 213
Structura sistemului de operare 213
Funcțiile principale ale sistemului de operare. 214
Fazele de pregătire ale unui program 215
Particularitățile sistemului de operare pentru calculatoarele de proces . . . 217
— *Programul monitor central* 221
— *Programul supervizor timp real* 222
— *Macroinstrucțiuni* 225
— *Programul de gestiune a liniilor* 229
— *Programe de diagnostic* 230
Exemple de sisteme de operare 232
— *Calculatorul FELIX C-32P* 232
— *Calculatorul IBM—1800* 236
Moduri de tratare a lucrărilor utilizate de calculatoarele de proces 236
— *Tratarea secvențială* 236
— *Multiprogramarea* 237
— *Tratarea conversațională* 238
— *Prelucrarea cu acces multiplu* 239
— *Prelucrarea în timp real* 241
12.2. Programe de aplicație 242
Tipuri de programe de aplicație. 243
— *Programe principale* 243

Scanned with OKEN Scanner

— *Programe parțiale* 244
— *Subprograme* 245
— *Transferări de date* 246
Sisteme de programare de aplicație 247
— *Părțile componente ale unui sistem de programe* 247
— *Sisteme de programe standard* 248
— *Generarea sistemului de programe* 250
12.3. Limbaje de programare pentru calculatoarele de proces 251
Cerințele impuse unui limbaj de proces 252
Limbaje de programare utilizate 253
— *Limbaje specifice aplicației* 254
— *Limbaje universale* 254
Exemple 255
— *Limbajul PROSPRO* 255
— *Program scris în limbajul AUTRAN* 260
Bibliografie 262

Scanned with OKEN Scanner

Capitolul 10

UTILIZAREA SISTEMELOR DE CALCUL LA CONDUCEREA PROCESELOR INDUSTRIALE

În acest capitol sînt prezentate principiile de bază şi cele mai uzuale procedee de control-reglare şi optimizare automată a proceselor tehnologice, cu scopul de a introduce cititorul în sfera acestor probleme şi de a-l iniţia asupra modului de abordare şi rezolvare a lor pe baza utilizării calculatoarelor electronice.

Organizarea capitolului s-a realizat pe baza principiului tratării conţinutului de la simplu la complex. Acolo unde volumul lucrării a permis, au fost inserate şi exemple sau exerciţii mai simple, urmînd ca alte aplicaţii mai ample ale chestiunilor teoretice tratate aici să fie prezentate în următoarele capitole.

Un loc important în acest capitol îl ocupă problematica referitoare la modelare-identificare, procedee care stau la baza teoriei şi practicii conducerii automate optimale a proceselor tehnologice cu mijloacele sistemelor de prelucrare automată a datelor.

10.1. PRINCIPII FUNDAMENTALE

Realizarea practică a oricărui proces tehnologic presupune, pe lîngă existenţa condiţiilor materiale necesare executării acestuia (instalaţii tehnologice, materie primă, energie etc.), şi desfăşurarea unei activităţi de control, reglare şi conducere a procesului.

Necesitatea unei astfel de activităţi este impusă de faptul că, în general, desfăşurarea proceselor tehnologice în concordanţă cu prescripţiile şi scopul pentru care au fost concepute implică o coordonare şi o corelare a diverselor operaţii fizice asupra unor elemente ale instalaţiilor tehnologice, precum şi de faptul că asupra proceselor tehnologice acţionează, de regulă, o serie de factori perturbatori cum ar fi acţiunea mediului ambiant, interacţiunea cu alte procese, accidente ş.a., care ar putea duce la o desfăşurare nedorită sau chiar periculoasă a proceselor.

O posibilitate practică, deosebit de eficientă, de supraveghere şi conducere continuă a proceselor constă în utilizarea calculatoarelor automate. Acestea, dotate cu sisteme de programe corespunzătoare procesului con-

Scanned with OKEN Scanner

trolat, „captează" prin „senzori" date cu privire la starea reală a procesului și instalației tehnologice de condus, le prelucrează conform unor modele și algoritmi stabiliți aprioric și furnizează (în timp real*) datele necesare procesului de decizie, reglînd prin semnalele „emise" anumiți parametri ai procesului. Aceste sisteme de calcul, utilizate la conducerea automată a proceselor, poartă denumirea de *calculatoare de proces*, care — în principiu — sînt calculatoare specializate (deși majoritatea pot fi utilizate și în mod tradițional, ca niște calculatoare universale).

În cadrul acestui subcapitol vom prezenta principiile de bază ale utilizării calculatoarelor la conducerea proceselor tehnologice.

Vom începe prin a comenta succint cîteva din noțiunile generale ale teoriei conducerii automate a proceselor. Recomandăm cititorului să revadă și lucrarea [9].

Noțiuni generale

Prin *control* se înțelege urmărirea desfășurării unui proces tehnologic dintr-o instalație, dintr-un agregat ș.a.m.d., prin măsurarea directă sau determinarea indirectă a variabilelor ce caracterizează starea sau desfășurarea acestuia și, eventual, prin semnalizarea optică și/sau acustică a depășirii anumitor limite de către variabilele de stare.

O astfel de operație se realizează cu ajutorul aparatelor de măsurat și eventul de semnalizare, iar informația primită pe această cale servește și este folosită la luarea deciziilor de conducere a procesului tehnologic în cauză.

Prin *reglare* se înțelege ansamblul de acțiuni prin care o variabilă — variabila reglată — a unui proces în desfășurare este menținută permanent în concordanță cu o altă variabilă — variabila de referință — pe baza măsurării valorii curente a variabilei reglate, a comparării ei cu valoarea de referință pentru a depista abaterea, și a acționării de așa manieră asupra procesului încît diferența dintre ele (abaterea) să fie în permanență înlăturată.

Aceste operații pot fi îndeplinite de către om, care în acest scop folosește diverse dispozitive și aparate ca: traductoare, convertoare, elemente de execuție, dispozitive de calcul etc. sau, mai bine, pot fi îndeplinite în întregime de aparate și dispozitive dintre care rolul de seamă revine unui *regulator automat*. Acesta îndeplinește funcția de comparare a celor două variabile pentru a determina abaterea și funcția de elaborare a semnalului de comandă, menit să ducă la lichidarea acestei abateri.

Prin *conducere* se înțelege o succesiune de acțiuni premeditate asupra unor elemente ale unui proces care sînt determinate ca rezultat al prelucrării informației privind desfășurarea procesului în cauză și executate după un anumit program, dinainte stabilit, în concordanță cu particularitățile și scopul pentru care a fost elaborat procesul. În esență este vorba de două genuri de acțiuni și anume:

---

* Noțiunea de timp real trebuie înțeleasă ca un concept care se aplică numai elementelor din configurația unui sistem de prelucrare a informației, ce sînt incluse în procese în care datele și rezultatele se referă la evenimente în „curs".

Scanned with OKEN Scanner

— prelucrarea informației referitoare la proces în vederea determinării deciziilor și acțiunii de conducere a acestuia, avînd la bază o anumită tactică și strategie;

— organizarea executării și realizarea acțiunilor de conducere adoptate, care se materializează sub forma unui flux de comenzi asupra procesului, avînd ca efect dirijarea acestuia.

Conducerea face parte din însăși esența oricărui proces tehnologic. Ea se poate efectua sub modalități foarte variate și este cu atît mai complexă cu cît procesul este mai amplu, mai rapid și are mai multe posibilități de desfășurare. Din marea varietate a acestor modalități și forme urmează să se aleagă acea conducere care se dovedește a fi cea mai potrivită particularităților și exigențelor procesului condus.

Așa de exemplu, dacă ne referim la un proces de stabilizare a unei variabile, adoptarea unei simple reglări poate satisface pe deplin cerințele acestui proces, pe cînd dacă ne referim la conducerea unei instalații tehnologice complexe, numai simpla reglare a unor variabile nu mai este suficientă, ci trebuie să recurgem la optimizarea unor raporturi între diversele variabile ale procesului pentru a obține un profit maxim.

În ultimă instanță, toate acțiunile de conducere din cadrul unui sistem trebuie astfel organizate încît să conducă la optimizarea activității ce se desfășoară în sistem, adică să asigure o eficiență maximă a procesului condus. *Optimizarea* reprezintă deci ansamblul măsurilor tehnico-organizatorice și economice prin care se asigură cel mai înalt (exigent) nivel de eficiență, adică acea conducere care asigură cele mai convenabile performanțe ce pot fi obținute din partea procesului condus, exprimate cantitativ printr-o funcție de performanță (care în majoritatea cazurilor reprezintă valori bănești).

Cea mai importantă caracteristică a activității de conducere, sesizată încă din 1948 de Norbert Wiener, constă în aceea că această activitate are loc în sisteme alcătuite din diverse elemente fizice la care participă și oameni pentru a transmite și a prelucra informația în vederea elaborării deciziilor și a comenzilor de conducere. Prin urmare, problemele de conducere se reduc în esență la probleme de transmitere și prelucrare a informației și la executarea comenzilor.

Această caracteristică esențială, ca și altele care derivă de aici, au deschis posibilitatea tratării unitare a conducerii sub aspectul ei esențial și au dus la conturarea unor discipline de sine stătătoare cum ar fi: cibernetica, automatica, informatica ș.a., care au la bază metoda analizei fluxului de informație.

Conform acestei metode, sistemele de conducere sînt divizate (fizic sau imaginar) în elemente componente cărora le sînt asociate anumite modele, exprimate prin ecuații sau funcții de transfer ale căror variabile exprimă informație, făcînd abstracție de natura fizică a elementelor respective. Elementele componente și sistemele de conducere sînt tratate din punctul de vedere informațional făcînd abstracție de natura fizică a variabilelor sistemului: temperatura, debitul, prețul ș.a.

În sistemele de conducere, informația și/sau comenzile se transpun pe semnale purtătoare de informație, se prelucrează în anumite elemente ale acestora și se transmit de la un element component la altul într-un

Scanned with OKEN Scanner

mod care în unele privinţe se aseamănă cu modul în care se prelucrează şi se transmite materia şi/sau energia în cadrul unei instalaţii tehnologice. Fluxul de informaţie şi/sau comenzi nu se substitue şi nu trebuie confundat cu fluxul de materie şi/sau energie, deşi în unele privinţe organigrama fluxului informaţional se aseamănă cu schema instalaţiei tehnologice.

Principiul acţiunii după abatere

Schema generală simplificată (organigrama) unui *sistem de conducere cu acţiune după abatere* (eroare) sau, cum se mai spune, *cu legătură inversă* (care este tipul fundamental de sistem de conducere din aproape totalitatea sistemelor de conducere) este prezentat în figura 10.1. Un astfel de sistem are două părţi componente principale:

— *partea condusă*, adică procesul tehnologic condus, *PC*, care se desfăşoară într-o instalaţie, maşină, agregat etc.;

— *partea conducătoare*, *MC*, alcătuită din totalitatea mijloacelor fizice cu ajutorul cărora se obţine informaţia necesară asupra desfăşurării procesului condus, se prelucrează această informaţie după un program dinainte stabilit şi se execută comenzi asupra procesului în vederea conducerii acestuia.

Între cele două părţi componente există o legătură directă, prin care se transmit comenzi de la partea conducătoare către partea condusă (proces) şi o legătură inversă (de la partea condusă către partea conducătoare), prin care se transmite informaţia necesară asupra desfăşurării procesului condus.

Existenţa legăturii inverse conferă acestor sisteme calitatea de autosesizare şi autocorecţie a propriilor erori, datorită faptului că ele analizează şi compară în permanenţă starea curentă a procesului condus cu o stare de referinţă şi acţionează permanent asupra procesului astfel încît neconcordanţa dintre cele două stări să fie lichidată.

Orice acţiune bazată pe acest principiu presupune constatarea unei discordanţe dintre starea curentă a procesului condus şi o stare de referinţă memorată; prin urmare, în astfel de sisteme, acţiunea de conducere intervine după apariţia şi constatarea unor deosebiri. Rezultă că acestor sisteme le este inerentă o oarecare întîrziere în transmiterea comenzilor. Se urmăreşte însă ca această întîrziere să fie cît mai mică posibil pentru a evita neajunsurile ce le-ar putea cauza.

Un sistem cu acţiune după abatere este capabil să anihileze efectul nefavorabil al oricărei perturbaţii asupra procesului; de aceea, astfel de sisteme sînt recomandabile pentru conducerea proceselor asupra cărora acţionează un număr mare de factori perturbatori şi unde interdependenţa dintre variabilele de stare nu este suficient de bine cunoscută.

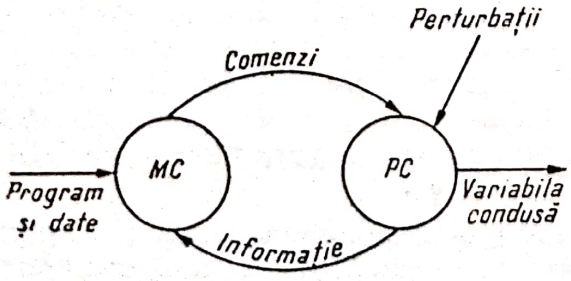


Fig. 10.1.

La astfel de sisteme cunoaşterea apriorică a numărului factorilor perturbatori şi a efectelor acestora asupra procesului condus nu este esenţia-

lă, deoarece acţiunea de conducere este bazată pe efectul deja creat, pe efectul constatat, fără să mai aibă vreo importanţă cauza care l-a creat.

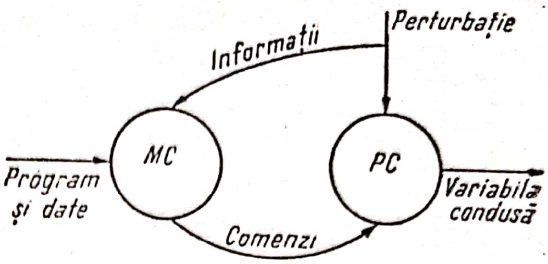


Fig. 10.2.

Principiul acţiunii după abatere şi sistemele de conducere bazate pe acest principiu au cea mai mare aplicabilitate în practică, deoarece asupra proceselor conduse acţionează în general un mare număr de factori perturbatori, iar dependenţa cauzală dintre variabilele acestor procese nu poate fi pe deplin cunoscută sub aspect cantitativ.

Principiul acţiunii după perturbaţie

În anumite cazuri, în care modelul sistemului (procesului condus) este bine cunoscut, poate fi aplicat principiul acţiunii după perturbaţie care constă în elaborarea şi realizarea unei acţiuni de înlăturare a efectului nedorit al unei perturbaţii pe baza cunoaşterii în permanenţă a perturbaţiei şi a dependenţei dintre aceasta şi efectul ei asupra procesului condus (fig. 10.2).

În astfel de sisteme se poate elimina efectul nedorit al unui factor perturbator datorită faptului că odată cu apariţia acestuia are loc determinarea apriorică a stării viitoare, cauzată de perturbaţie, o comparaţie a acestei stări cu o stare de referinţă şi realizarea pe această bază a unei comenzi menite să preîntîmpine orice abatere a sistemului de la starea de referinţă. Deci, sub acest aspect, sistemele cu acţiune după perturbaţie sînt sisteme cu acţiune după o abatere presupusă aprioric.

În sistemele cu acţiune după perturbaţie nu există o legătură inversă între partea condusă şi partea conducătoare, care să permită analiza efectivă a stării curente a procesului condus în vederea unei eventuale corecţii; acestea sînt deci sisteme cu circuit deschis. Aici analiza procesului şi comparaţia cu starea de referinţă se face aprioric, adică pe legi sau presupuneri prestabilite care nu întotdeauna concordă cu realitatea.

De asemenea, dependenţa cantitativă dintre cauză şi efect, adică dintre perturbaţii şi efectele lor asupra procesului condus, este foarte greu de stabilit mai ales la procesele complexe unde între variabilele acestora există corelaţii aproape imposibil de modelat, pentru că de cele mai multe ori au un caracter dinamic. Din această cauză conducerea după acest principiu poate fi adoptată numai pentru procese bine cunoscute (cu modele matematice complete şi precise), cu un număr relativ mic de factori perturbatori.

Principiul acţiunii combinate

În cadrul unui sistem de conducere poate fi adoptat atît principiul acţiunii după abatere cît şi principiul acţiunii după perturbaţie, realizîndu-se astfel un *sistem combinat* (fig. 10.3), adică un *sistem cu acţiune combinată* în care se întrunesc avantajele celor două tipuri de conducere prezentate mai înainte.

Scanned with OKEN Scanner

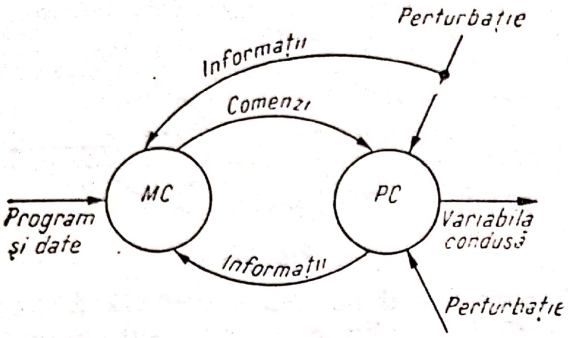


Fig. 10.3.

Într-un astfel de sistem combinat se introduce, de obicei, acțiune numai după principalii factori perturbatori (pentru a preîntîmpina abaterile nedorite ale procesului condus de la starea de referință ce ar putea fi cauzate de aceștia), urmînd ca abaterile provocate de ceilalți factori perturbatori ca și eventualele erori ale subsistemelor cu acțiune după perturbație să fie lichidate de subsistemul cu acțiune după abatere.

În ultima vreme, se folosesc tot mai frecvent sisteme cu acțiune preponderentă după perturbație, adică așa-numitele *sisteme predictive*. Pe măsura cunoașterii aprofundate a proceselor, reglarea predictivă s-a dovedit a fi deosebit de avantajoasă, îndeosebi în acele aplicații la care există posibilitatea analizei continue și cu mijloace economice a variabilelor de intrare și a efectului potențial al acestora asupra stării și desfășurării procesului condus.

Structura generală a sistemelor de conducere

În general orice proces (și deci și procesul de conducere) se găsește într-o strînsă dependență cu alte procese, cu care se influențează reciproc, și este supus influenței unor factori aleatori independenți față de proces.

Pentru a facilita conducerea unui astfel de proces se recurge la delimitarea, din cadrul marelui ansamblu al naturii, a unui sistem sau subsistem de conducere ținînd cont de legăturile acestuia cu restul ansamblului. În acest fel, toate sistemele sau subsistemele de conducere se pot aranja după gradul lor de subordonare; se pot deci ierarhiza pe două sau mai multe nivele în funcție de ansamblul luat în considerație (figura 10.4 în care: PC — proces condus; MC — mijloace de conducere; PD — programe și date; CD — comenzi sau decizii; I — informație și P — perturbație).

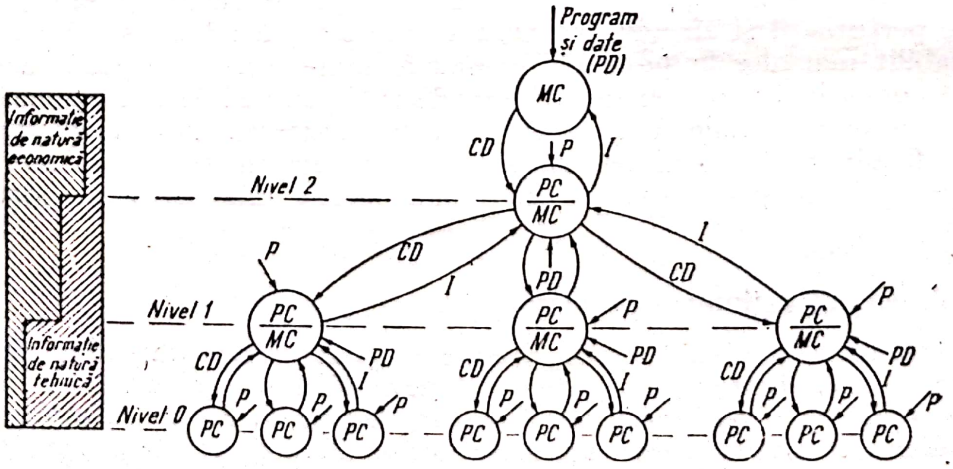


Fig. 10.4.

În cadrul unui ansamblu (sistem unitar integrat) informaţia „circulă" în ambele sensuri (de jos în sus şi de sus în jos), însă la graniţa dintre un nivel şi altul are loc o transformare, o „condensare" şi o selecţie a informaţiei, astfel că numai o parte din aceasta circulă spre nivelul superior. De regulă, prin selecţia menţionată se face ca ponderea informaţiei de natură economică să crească spre nivelele superioare şi deci informaţia de natură tehnică să aibă ponderea cea mai mare la nivelul cel mai scăzut — nivelul 0.

De asemenea, este important de semnalat şi faptul că la nivele intermediare partea condusă a subsistemului ierarhic superior constituie în acelaşi timp şi partea conducătoare pentru subsistemul ierarhic inferior (din „aval").

Ierarhizarea conducerii este aplicată în toate compartimentele vieţii economice şi sociale, deoarece toate fenomenele economice şi sociale se găsesc într-o strînsă interdependenţă pe plan naţional şi internaţional.

În sprijinul acestei idei, dăm ca exemplu conducerea unei instalaţii de polietilenă din R.S. România care nu poate face abstracţie de producţia de polietilenă pe plan naţional sau chiar mondial, pentru că altfel ar putea să se producă sortimente care sînt în exces pe piaţă şi deci să nu se mai vîndă sau să nu se profite de o conjunctură favorabilă (cereri) pe piaţa mondială (probleme de marketing).

Nivelele de la care ar urma să se exercite conducerea sînt determinate de modul de organizare a vieţii economice şi social-politice pe plan naţional şi internaţional. Pentru ţara noastră conducerea se exercită la cel puţin următoarele nivele: guvern-ministere-centrale-întreprinderi-secţii-instalaţii.

10.2. METODE ŞI MIJLOACE DE CONDUCERE A PROCESELOR TEHNOLOGICE

Funcţiile şi configuraţia sistemelor de conducere au evoluat în decursul timpului, fiind rezultatul progresului continuu al gîndirii ştiinţifico-tehnice, al tehnologiei în general, al gradului de cunoaştere şi al posibilităţilor de materializare a gîndirii progresiste. Se poate spune că gradul de dezvoltare economică şi socială a unei colectivităţi umane se reflectă şi în gradul de automatizare a activităţilor acestei colectivităţi, în configuraţiile sistemelor de conducere adoptate.

În cele ce urmează sînt descrise cîteva configuraţii tipice, care s-au folosit, se folosesc sau se vor folosi, în ordinea istorică a apariţiei şi folosirii lor, respectînd principiul tratării de la simplu la complex.

Conducerea cu mijloace tradiţionale

Conducerea directă de către om. În figura 10.5 este prezentat un sistem de conducere, unde rolul principal revine omului (unui „operator uman"), care este situat pe legătura inversă a acestui sistem. Pentru a realiza conducerea, omul se foloseşte de

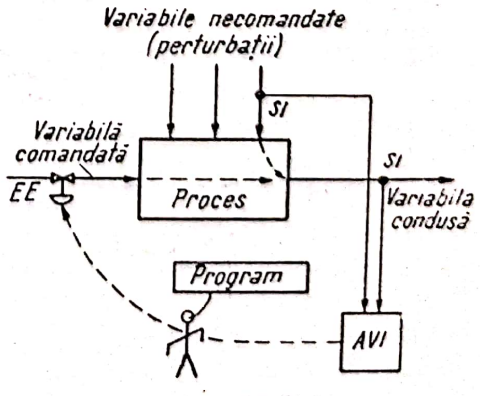


Fig. 10.5.

mai multe surse de informaţii *(SI)*, de unul sau mai multe aparate de măsurat şi vizualizare a informaţiei privind variabilele caracteristice ale procesului condus *(AVI)*, precum şi de unul sau mai multe elemente de execuţie *(EE)* prin intermediul cărora acţionează asupra procesului în vederea menţinerii variabilelor de bază ale acestuia între limitele dorite sau admisibile.

Acest mod de conducere se situează la limita inferioară a evoluţiei sistemelor de conducere, fiind caracterizat prin aceea că omului îi revine sarcina prelucrării informaţiei şi executării comenzilor după un program prestabilit.

După caz, el poate acţiona fie numai în funcţie de abatere, fie numai de perturbaţii, precum şi combinat.

Conducerea cu ajutorul regulatoarelor. Un prim pas în evoluţia sistemelor de conducere l-a constituit introducerea regulatoarelor ca elemente de prelucrare a informaţiei şi de comandă, care a permis eliminarea omului din aceste funcţii. Schema simplificată a unui sistem de reglare automată locală este prezentată în figura 10.6, de unde se vede că omului îi revine numai sarcina de a determina în permanenţă valorile de referinţă ale regulatoarelor şi de a le introduce în regulatoare în funcţie de condiţiile de lucru ale procesului condus, încadrat într-un ansamblu.

Pentru exercitarea funcţiilor sale, personalul de conducere primeşte informaţia necesară privind procesul condus de la sursele de informaţie *(SI)* prin intermediul aparatelor de vizualizare a informaţiei *(AVI)*, prelucrează această informaţie pentru a determina valorile de referinţă ale regulatoarelor şi execută diverse acţiuni de conducere cum ar fi, de exemplu, pornirea sau oprirea unei pompe, închiderea sau deschiderea unui robinet, modificarea valorii de referinţă a unui regulator ş.a.

Reglarea automată locală se utilizează în special pentru procesele relativ simple şi izolate, adică fără interconexiuni puternice cu alte procese ce fac parte din acelaşi sistem sau din alte sisteme.

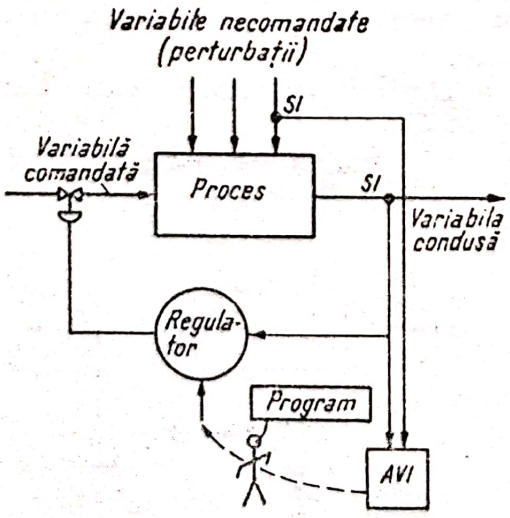


Fig. 10.6.

Conducerea cu ajutorul instalaţiilor de control şi reglare centralizate. Conducerea efectivă a proceselor tehnologice dintr-o instalaţie industrială se realizează, de regulă, dintr-un singur loc central — din camera dispecerului. Aici sînt amplasate aproape toate aparatele pentru vizualizarea informaţiei provenite din instalaţie, precum şi aproape toate regulatoarele şi dispozitivele de comandă a diverselor elemente de execuţie aflate în instalaţie.

În figura 10.7 este prezentată schematic o instalaţie de conducere centralizată formată din aparate de măsurat, vizualizat şi/sau semnalizat valorile fie-

Scanned with OKEN Scanner

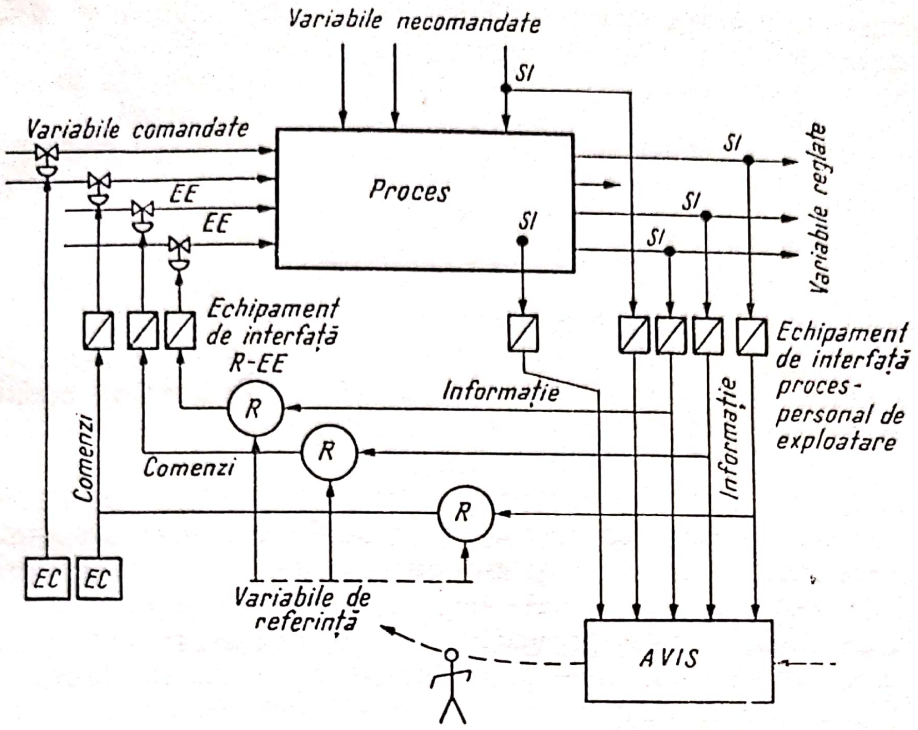


Fig. 10.7.

cărei variabile importante a procesului condus *(AVIS)*, care primește date de la sursele de informație *(SI)*, din regulatoarele de proces *(R)*, care acționează asupra elementelor de execuție *(EE)* ale instalației tehnologice, precum și din aparatura tip interfață proces-regulator. Evident, în instalație există unul sau mai mulți operatori însărcinați cu conducerea procesului.

În funcție de numărul și gabaritele aparatelor menționate mai înainte, acestea se montează pe panouri sau pe pupitre de așa manieră încît să se poată identifica cu ușurință toate variabilele controlate sau reglate. În acest scop, se recurge la amplasarea acestora pe o schemă sinoptică a instalației tehnologice și la atașarea unor inscripții de identificare lîngă fiecare aparat.

Necesitatea exercitării controlului și reglării centralizate a proceselor tehnologice moderne a stimulat progresul în următoarele domenii ale construcției de aparatură pentru automatizare:

— realizarea de aparatură cu gabarite reduse;
— dezvoltarea producției de aparatură unificată;
— realizarea echipamentului de interfață proces-element de conducere.

Aproape în totalitatea ei aparatura de măsurat și automatizare care se fabrică în prezent corespunde unor norme ale unor sisteme unificate, elaborate de principalii producători de asemenea mijloace tehnice, care realizează prin acestea tipizarea (standardizarea) aparatelor sub aspect constructiv și funcțional, punînd accent pe modularizarea acestora și pe reducerea gabaritelor, precum și standardizarea în ce privește natura și limitele de variație ale semnalelor purtătoare de informație (de exemplu: curent electric între limitele 2 și 10 mA, tensiune electrică de la 0 la 6 V ș.a.).

Scanned with OKEN Scanner

Tipizarea şi minaturizarea aparaturii unificate asigură, pe lîngă avantaje economice, o utilizare aproape universală a acestei aparaturi (cu excepţia traductoarelor primare) şi toate avantajele interşarjabilităţii, atît în fabricaţie cît şi în exploatare.

Pentru a face transferul de informaţie de la sursele de informaţie (traductoare), care sînt specifice variabilei măsurate, la aparatele de vizualizare a informaţiei sau la aparatele de reglare unificate, ca şi pentru transmiterea comenzilor de la aparatura de comandă unificată la diversele elemente de execuţie din cadrul instalaţiei tehnologice, se recurge la folosirea unui echipament de legătură, constituit în principal din diverse tipuri de convertoare şi adaptoare de semnal, de asemenea, şi ele unificate. Deoarece aceste elemente fac legătura între proces şi un element de comandă, ele sînt denumite *echipamente de interfaţă* proces-element de conducere.

În sistemul din figura 10.7, conducerea procesului revine personalului de exploatare. În cazul unor procese relativ simple şi lente acest personal poate exercita cu succes această funcţie. Există însă procese tehnologice moderne foarte complexe şi rapide la care, cu toată experienţa sa, personalul de exploatare nu poate cuprinde, asimila şi interpreta în timp util tot fluxul de informaţie provenind din instalaţie, nu are posibilitatea şi timpul necesar să prelucreze şi să interpreteze această informaţie. Deci el nu poate executa la timp toate manevrele (comenzile) necesare pentru dirijarea procesului, şi poate comite numeroase greşeli în aceste operaţii.

Toate aceste neajunsuri pledează în favoarea utilizării unor mijloace moderne de control, reglare şi prelucrare automată a informaţiei care să scutească personalul de exploatare, măcar parţial, de această activitate de rutină foarte obositoare.

Conducerea cu ajutorul calculatoarelor de proces

În actuala etapă de dezvoltare a tehnicii şi tehnologiei, procesele tehnologice se caracterizează printr-un număr mare de variabile, prin viteze de desfăşurare mari, prin interdependenţe interne şi externe puternice, prin exigenţe tot mai mari şi numeroase restricţii privind producţia realizată, siguranţa în funcţionare a instalaţiei tehnologice ş.a.

Faţă de această situaţie, posibilităţile organismului uman de a colecta rapid un mare volum de informaţie, de a prelucra şi interpreta această informaţie în vederea determinării deciziilor şi comenzilor ce se impun şi de a executa manevrele de conducere necesare se dovedesc a fi cu totul nesatisfăcătoare şi prea lente pentru a se acţiona în timp util.

Conducerea optimă a unor astfel de procese complexe este o problemă dintre cele mai dificile. Personalul de conducere a acestor procese este copleşit de marea cantitate de informaţie, de multitudinea soluţiilor posibile pentru multitudinea stărilor posibile în desfăşurarea procesului, de rapiditatea cu care trebuie să ia şi să execute deciziile de conducere ; el este în permanenţă în „criză de timp" şi confruntat cu multe dileme, de cele mai multe ori fiind nevoit să recurgă la soluţii bazate pe intuiţie.

Existenţa relaţiilor neliniare dintre numeroasele variabile ale procesului, a numeroaselor restricţii cu privire la limitele lor de variaţie ca şi întîrzierile (inerţia) cu care procesul răspunde la anumite comenzi ridică gradul de dificultate a conducerii.

Scanned with OKEN Scanner

Așa de exemplu, se întîmplă adesea ca modificarea valorii unei variabile, efectuată de personalul de exploatare în virtutea unei decizii de conducere, să aducă după sine modificarea, cu oarecare întîrziere, a multor altor variabile și ieșirea dintre limitele admisibile a unora dintre ele. Aceasta face necesară o nouă modificare a variabilei respective pentru a reîncadra între limitele admisibile celelalte variabile. Dar și această modificare ar putea avea efecte similare mai ales dacă procesul este supus acțiunii unor perturbații. Prin urmare, astfel de efecte se pot ține lanț. Pentru a le evita este necesar ca orice acțiune de comandă să fie elaborată pe baza analizei exacte și complete a întregului sistem, ceea ce nu se poate face fără utilizarea mijloacelor moderne de prelucrare automată a informației, adică a utilizării calculatoarelor ca instrumente de prelucrare automată a informației în vederea determinării rapide și cu precizie a comenzilor necesare desfășurării optime a procesului condus, și eventual ca instrument de comandă directă a acestuia.

Calculatoarele pot fi folosite în cele mai felurite moduri în funcție de gradul de cunoaștere a proceselor conduse, de experiența acumulată, de mijloacele tehnice de automatizare de care se dispune, precum și de alți factori. În cele ce urmează sînt prezentate cîteva moduri tipice de folosire a acestor mijloace.

Conducerea cu ajutorul mașinilor de control și reglare centralizate. O soluție de început în ce privește promovarea tehnicii moderne de calcul numeric în controlul și reglarea nemijlocită a proceselor tehnologice complexe o constituie realizarea și utilizarea *Mașinilor de Control și Reglare automată Centralizată (MCRC)*, care beneficiază de avantajele evidente ale prelucrării informației sub formă numerică față de prelucrarea acesteia sub formă analogică: precizie mult mai mare, memorare și programare mult mai ușoare, facilități de calcul mult mai mari ș.a.

MCRC pot înlocui cu succes zeci sau sute de aparate de vizualizare a datelor, precum și zeci de regulatoare convenționale (tradiționale), printr-un singur sistem de vizualizare a informației (imprimantă și/sau indicator) și printr-un singur regulator multiplu avînd performanțe ridicate, care este conectat succesiv la fiecare variabilă controlată sau reglată.

O astfel de mașină poate fi, de asemenea, programată să îndeplinească și funcția de semnalizare optică și/sau acustică a depășirilor de către anumite variabile a unor valori limită, precum și diverse funcții de calcul și înregistrare sistematică sau la cerere a unor variabile calculabile cum ar fi: randament, conținut caloric, viteze de reacție, volumul producției și consumurile pe o anumită perioadă de timp, stocurile ș.a.

În figura 10.8 se sugerează modul de utilizare a unei *MCRC* în conducerea unui proces tehnologic (în această figură s-au folosit notațiile: *CAN* — convertor analog-numeric, *CNA* — convertor numeric — analogic și *AVIS* — aparate de vizualizare a informației și semnalizare). Din această figură se remarcă faptul că între proces și echipamentul de interfață proces — *MCRC* există numeroase linii de transmitere a informației și comenzilor analogice, pe cînd între echipamentul de interfață și *MCRC* nu există decît cîte o singură linie pentru fiecare sens prin care se transmit informația și comenzile sub formă numerică.

Scanned with OKEN Scanner

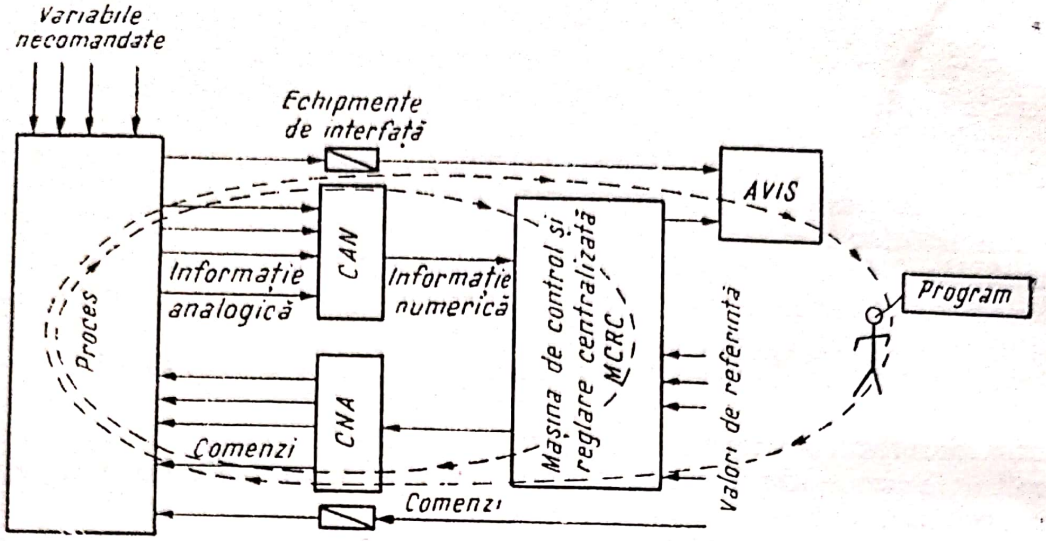


Fig. 10.8.

Rezultă deci că echipamentul de interfață proces — *MCRC* trebuie să aibă în componența sa, pe lîngă convertoarele și adaptoarele necesare transpunerii informației primare pe un semnal analogic standard, și un convertor analog-numeric *(CAN)* care să transforme valorile analogice în valori numerice cu care operează *MCRC*. Acest convertor unic, ca și *MCRC*-ul, se conectează pe rînd sau la cerere cu fiecare din sursele de informație. Apartenența sa la echipamentul de interfață sau la *MCRC* este o chestiune de convenție.

Rezultatele măsurărilor ca și rezultatele calculelor efectuate de *MCRC* sînt vizualizate în diverse coduri numerice sau alfanumerice de către mașini imprimante speciale, iar comenzile care urmează a fi executate de diverse elemente de execuție analogice sînt mai întîi convertite din cod numeric în semnale analogice folosind un convertor numeric — analogic *(CNA)*, apoi distribuite succesiv și adaptate la fiecare element de execuție.

Avînd la dispoziție o astfel de mașină cu un înalt grad de centralizare a datelor și a comenzilor, care poate realiza și funcția de semnalizare a situațiilor anormale, precum și unele funcții de prelucrare a unor date privind procesul condus, personalul de exploatare poate exercita mult mai ușor funcția de conducere. Atenția lui nu va mai fi distribuită asupra numeroaselor aparate de măsurat sau ocupată cu efectuarea unor calcule sau raționamente de rutină, ci va fi îndreptată asupra *MCRC*-ului care-i furnizează date sintetice despre procesul condus, pe baza cărora să ia ușor decizii de conducere.

Figura 10.8 sugerează ideea că cea mai mare parte a fluxului de informație provenind de la procesul condus parcurge un circuit care se închide prin *MCRC* și numai o parte din acest flux ajunge la personalul de conducere și se „închide" prin acesta.

Conducerea cu calculatoare în regim de consultant. Cea mai simplă formă de utilizare a calculatoarelor pentru conducerea proceselor tehnologice, care constituie în același timp și o etapă de început în vederea acumulării experienței necesare, este forma de utilizare în calitate de consultant

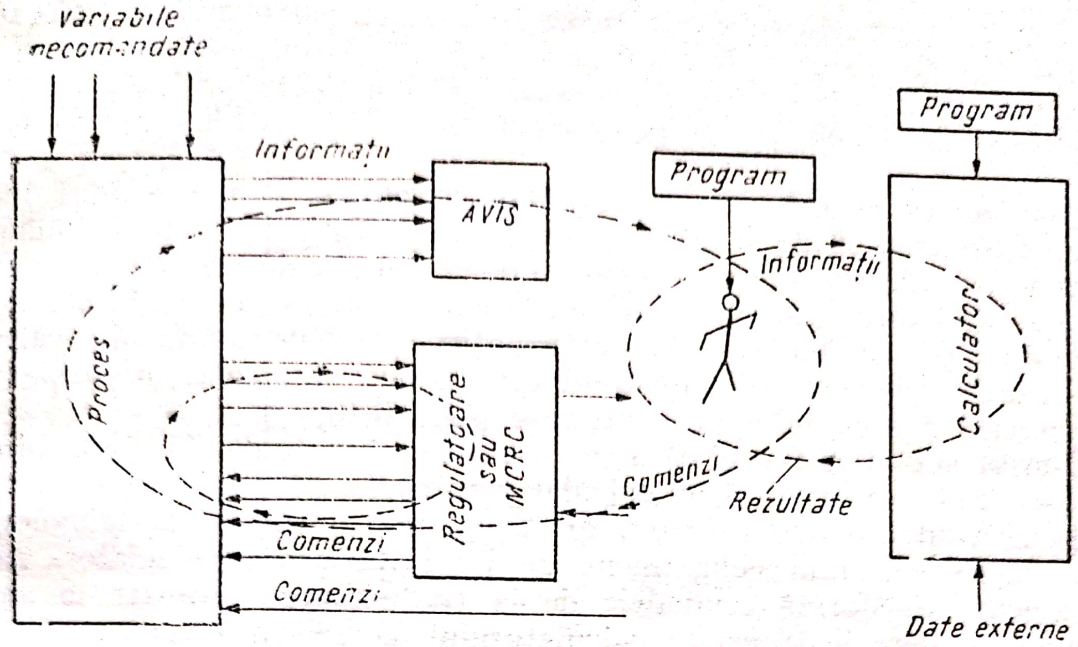


Fig. 10.9.

(„of line"). Sub această formă poate fi folosit orice fel de calculator universal în vederea prelucrării datelor primare provenind de la procesul condus pentru a furniza personalului de conducere a procesului date sintetice care să-l ajute la luarea deciziilor de conducere (fig. 10.9).

Condcerea în cadrul acestui sistem se realizează după cum urmează:

— principalele variabile tehnologice sînt controlate şi reglate în cadrul sistemului cu circuitul închis cel mai interior din care fac parte: instalaţia tehnologică, regulatoarele sau *MCRC* şi echipamentul de interfaţă proces — regulatoare;

— conducerea propriu-zisă a procesului în ansamblu se realizează în cadrul unui sistem ierarhic superior, în care intră ca parte componentă sistemul de control şi reglare menţionat mai înainte, personalul de conducere precum şi aparatura pentru vizualizarea informaţiei şi bineînţeles calculatorul;

— personalul de conducere analizează informaţia ce provine de la procesul condus şi de la conducerea ierarhic superioară, reţine ca atare o parte din această informaţie, iar pe cealaltă o transferă calculatorului pentru a o prelucra după un anumit program, dinainte stabilit, în vederea obţinerii unor rezultate sintetice;

— pe baza informaţiei preluate direct şi pe baza rezultatelor furnizate de calculator, acelaşi personal de conducere determină şi execută, după un anumit program, comenzile necesare asupra procesului, direct asupra unor elemente de execuţie sau prin modificarea valorilor de referinţă ale regulatoarelor.

Caracteristic pentru configuraţia din figura 10.9 este faptul că transferul de informaţie de la proces către calculator, ca şi executarea comenzilor de conducere asupra procesului sînt executate de către personalul de exploatare, acesta fiind o verigă prin care sistemul de conducere (de care aparţin regulatoarele sau *MCRC*-ul) se racordează cu sistemul de conducere ierarhic superior din care face parte calculatorul.

Scanned with OKEN Scanner

Intrucît, datorită reflexelor sale relativ lente, personalul de exploatare exercită această sarcină de transfer de informaţie cu o viteză relativ redusă, deci cu întîrzieri apreciabile, rezultă că o astfel de utilizare a calculatorului se recomandă numai pentru conducerea proceselor relativ lente.

Conducerea cu calculatoare conectate la proces. Pentru a mări operativitatea sistemului de conducere se poate adopta un sistem similar cu cel din figura 10.9 dar în care transferul de informaţie de la proces la calculator să se facă direct, nu prin intermediul personalului de conducere, bineînţeles după o prealabilă selecţie a informaţiei ce trebuie să ajungă la calculator. În acest fel, personalul de conducere rămîne intercalat numai pe legătura inversă a sistemului de conducere (fig. 10.10).

Merită subliniat faptul că în acest caz se foloseşte de obicei un *calculator numeric de proces*, adică un calculator specializat care primeşte informaţie de la traductoarele primare, aflate în cadrul instalaţiei tehnologice, prin intermediul unui echipament de interfaţă proces-calculator. Acest echipament transformă semnalele de la traductoarele primare în valori numerice pe care le transferă calculatorului, ce se conectează pe rînd şi foarte rapid la fiecare din traductoarele primare care-i furnizează informaţia, iar el, la rîndul său, furnizează personalului de conducere rezultatele sintetice ale prelucrării informaţiei primite. În rest, conducerea procesului se face aşa cum s-a arătat şi mai înainte, cu menţiunea că prin conectarea calculatorului la proces s-a format un al treilea flux de informaţie cu circuit închis, alcătuind un sistem ierarhic superior celorlalte două.

Progresele recente din domeniul tehnicii conducerii automate a proceselor tehnologice, materializate în realizarea diverselor tipuri de calculatoare de proces şi echipamente de interfaţă proces-calculator, fac posibilă extinderea pe o scară tot mai largă a conectării complete, bidirecţionale, a calculatoarelor la proces, eliminînd practic complet personalul de conducere din circuitul informaţional de bază al sistemelor de conducere.

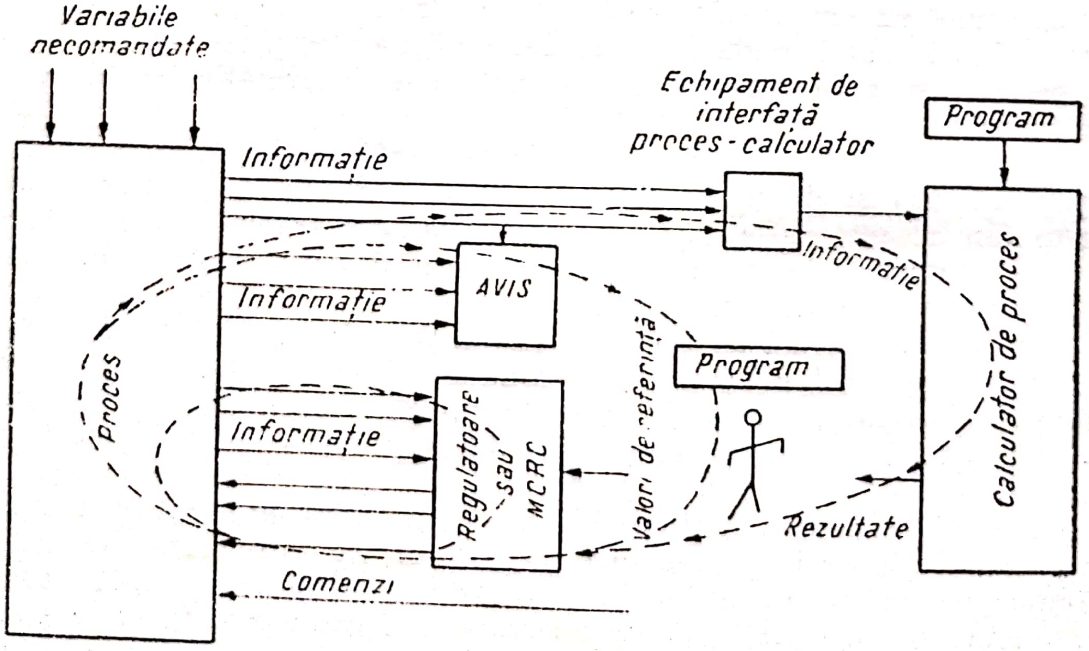


Fig. 10.10.

Scanned with OKEN Scanner

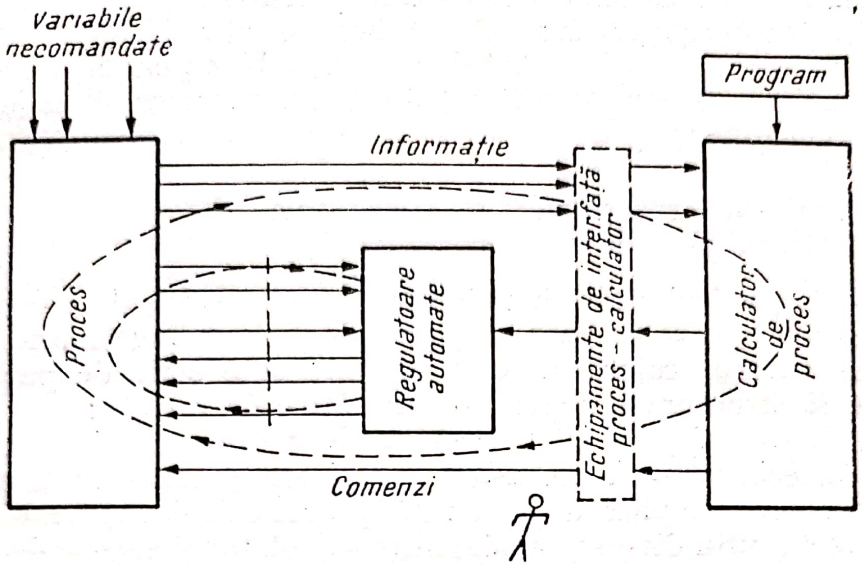


Fig. 10.11.

O configurație modernă a unui sistem de conducere care îmbină și adaptează tehnica tradițională a reglării automate cu noua tehnică de conducere automată de către calculator este prezentată simplificat în figura 10.11.

În cadrul acestui sistem, calculatorul primește informația provenind de la proces prin intermediul echipamentului de interfață, precum și informația de la sistemul ierarhic superior, prelucrează această informație în conformitate cu programul memorat și execută comenzile, ce rezultă din această prelucrare, asupra mărimilor de referință ale regulatoarelor sau asupra elementelor de execuție, prin intermediul echipamentului de interfață. În acest fel, circuitul de informație intermediar (care în figura 10.10 se închidea prin personalul de exploatare) se închide aici (v. fig. 10.11) prin calculatorul de proces, iar rolul personalului de conducere se reduce la programarea adecvată a calculatorului de proces și la supravegherea întregului sistem de conducere și a echipamentului tehnologic.

Cea mai modernă și eficientă soluție în automatizarea conducerii o constituie configurația prezentată simplificat în figura 10.12, unde este vorba de conducerea directă de către calculator a procesului tehnologic fără intermediul regulatoarelor, adică o *Conducere Numerică Directă (CND)*. În acest fel circuitul de informație care se închide prin calculator preia și informația care în variantele precedente era prelucrată în unul sau chiar două circuite închise de nivel ierarhic inferior.

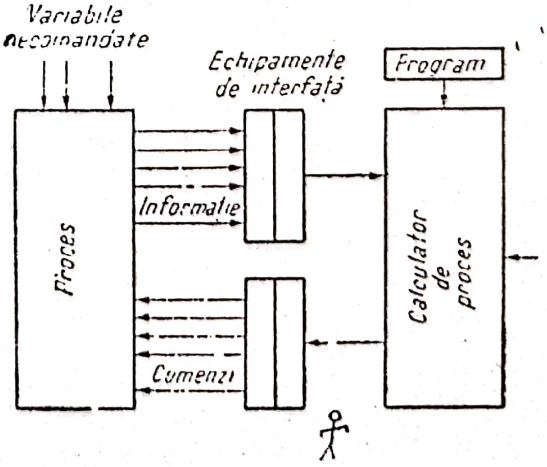


Fig. 10.12.

Realizarea unei astfel de conduceri presupune cunoaşterea perfectă a procesului tehnologic condus, adică posesia unui model matematic precis şi complet al procesului, un pachet de programe bine puse la punct, precum şi cele mai eficace măsuri şi mijloace de mărire a siguranţei în funcţionarea întregului echipament.

Particularităţile funcţionale şi structurale ale calculatoarelor de proces

Cerinţe şi particularităţi funcţionale. Calculatoarele de proces moderne sînt instalaţii complexe de înalt nivel tehnic care pe lîngă multe caracteristici comune tuturor calculatoarelor prezintă şi o serie de particularităţi funcţionale şi structurale impuse de următoarele cerinţe:

1° *Aptitudini şi posibilităţi multiple de a comunica direct cu procesul condus în ambele sensuri.* Această comunicare se face prin intermediul unui echipament complex de interfaţă proces-calculator, care mijloceşte legătura directă, prin diverse canale, între calculator şi sursele de informaţie (traductoare, sezizoare etc. cunoscute şi sub numele generic de senzori) şi între calculator şi elementele de execuţie din instalaţia tehnologică. Prin acest echipament calculatorul de proces se poate conecta ciclic, prin divizarea timpului, la zeci, sute şi chiar mii de surse de informaţie şi cu zeci sau sute de elemente de execuţie.

În mod normal conectarea calculatorului la fiecare sursă de informaţie sau element de execuţie se face pe rînd la intervale de timp dinainte stabilite, însă ea poate fi făcută şi altfel printr-un sistem de priorităţi stabilite prin program.

2° *Aptitudinea de a reacţiona în timp util la modificările variabilelor care definesc starea sau desfăşurarea procesului condus, adică de a lucra în „timp real".* În acest scop, prelucrarea informaţiei şi emiterea rezultatelor sau comenzilor trebuie să se facă suficient de rapid, astfel ca informaţia recepţională să nu-şi piardă valabilitatea pînă la executarea comenzilor ce se impun.

3° *Aptitudinea de a trece foarte rapid de la executarea unui program la executarea altui program cu un nivel de prioritate superior prin divizarea timpului de lucru.* Calculatoarele de proces trebuie să permită executarea simultană a mai multor programe sau subprograme, în sensul că un program aflat în lucru să poată fi întrerupt oricînd în favoarea executării altui program prioritar şi reluat cu uşurinţă după executarea acestuia din urmă.

4° *Existenţa unui sistem de programe specifice,* elaborate în concordanţă cu specificitatea procesului şi instalaţiei conduse sau adaptate la acestea.

5° *Facilităţi în realizarea unei comunicaţii între operatorii tehnologici şi proces prin intermediul calculatorului,* care să permită operatorilor accesul la informaţia furnizată de calculator şi la efectuarea unor comenzi cu sau fără ajutorul calculatorului.

6° *Posibilitatea de a realiza cu uşurinţă o configuraţie optimă prin ansamblarea unui număr oarecare de componente modulare* şi de a modifica structura iniţială în funcţie de nevoile şi pretenţiile utilizatorului, ţinînd cont de posibilele dezvoltări, atît ale instalaţiei tehnologice, cît şi ale calculatorului. Se cere deci o mare flexibilitate atît în ceea ce priveşte echipamentele cît şi programele.

Scanned with OKEN Scanner

7° *Posibilitatea interconectării cu alte echipamente de prelucrare a informației*, pentru a putea intra într-o asemenea rețea și de a accepta programe standard.

8° *Fiabilitate foarte înaltă*, de ordinul 99% pentru funcții de supraveghere și 99,9% pentru funcții de conducere și protecție, în condiții de exploatare industriale destul de grele, inclusiv stabilitate la fluctuații ale mediului ambiant. Se cere deci ca întregul echipament de prelucrare a informației să fie apt în a preveni, detecta și remedia deficiențele prin programe de diagnosticare și remediere și să dispună de suficiente resurse de alternativă care să fie folosite în caz de nevoie.

Toate aceste cerințe și particularități funcționale determină și structura generală a calculatorului de proces tratată succint în paragraful următor.

Structura generală a calculatoarelor de proces. În figura 10.13 este prezentată schema simplificată a configurației unui calculator de proces. Fără a intra în detalii de construcție și funcționare, vom evidenția cîteva din caracteristicile și particularitățile esențiale ale elementelor sale componente insistînd mai mult asupra aspectelor legate de exploatare.

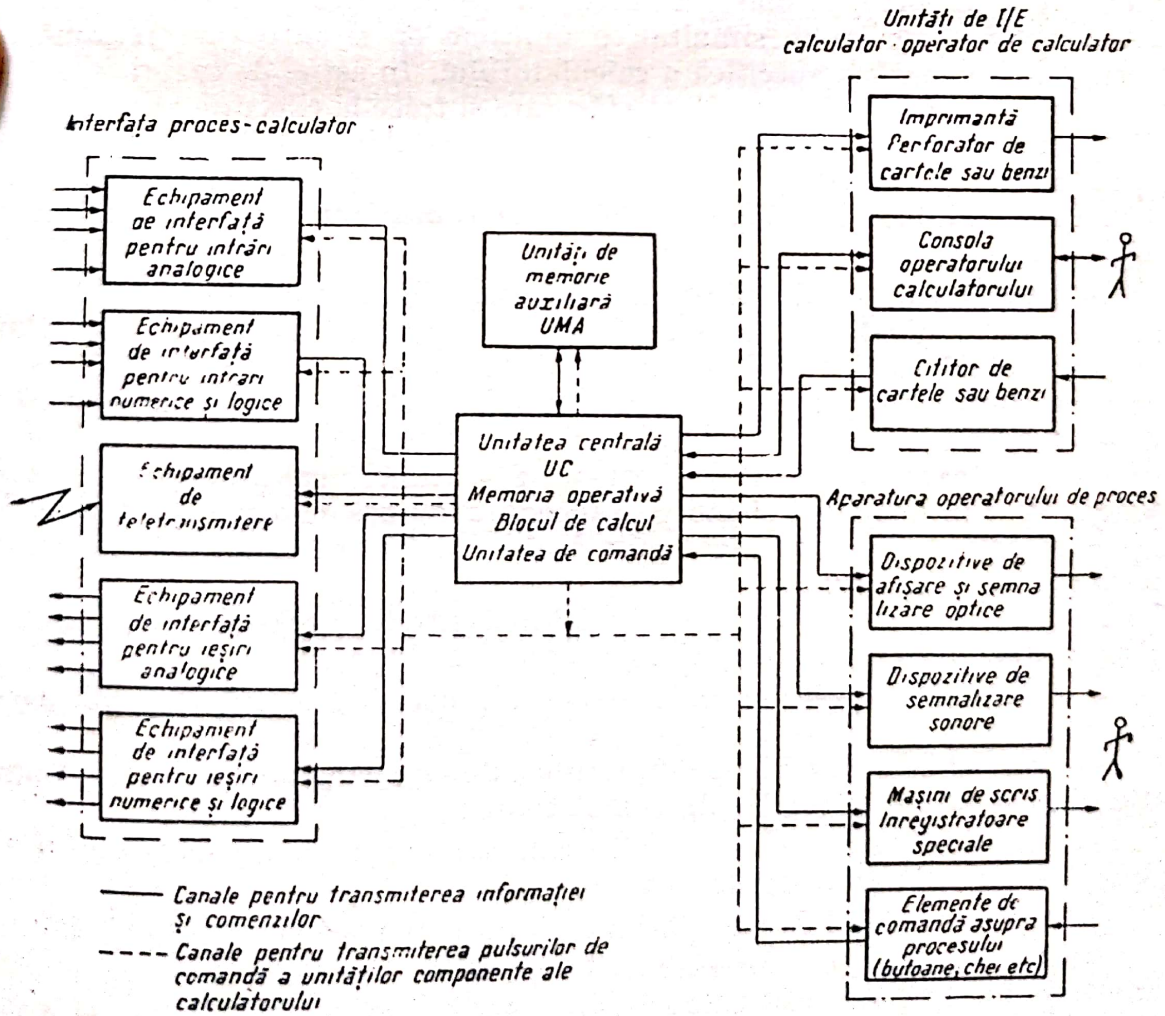


Fig. 10.13.

Scanned with OKEN Scanner

Dintre elementele componente remarcăm: unitatea centrală *UC*; unităţile de memorie auxiliară *UMA*; unităţile de interfaţă calculator-operator şi echipamentul de interfaţă proces-calculator.

Unitatea centrală nu se deosebeşte substanţial de unităţile centrale ale calculatoarelor numerice universale şi se caracterizează prin următoarele proprietăţi: lungimea şi structura cuvintelor; setul de instrucţiuni şi modul de adresare; modul de recepţie şi transfer de date şi sistemul de priorităţi.

De remarcat ar fi modalităţile mai variate în ceea ce priveşte recepţia şi transferul de date la calculatoarele de proces. Motivat de necesitatea unei comunicări imediate cu mai multe surse de informaţie, concomitent cu operaţiile de calcul, aceste calculatoare sînt capabile să recepţioneze şi să transmită date prin următoarele procedee:

— introducerea programată direct în *UC* pe baza instrucţiunilor din programul de exploatare, procedeu specific acestor calculatoare;

— introducerea prin intermediul registrelor tampon, unde informaţia este reţinută provizoriu pînă cînd *UC* sau o unitate periferică devine disponibilă să o primească şi să opereze asupra ei.

La calculatoarele de proces sistemul de priorităţi de execuţie şi întreruperi are o importanţă mare, deoarece în desfăşurarea proceselor tehnologice complexe pot apare simultan o mulţime de situaţii care reclamă o intervenţie imediată şi specifică a calculatorului. În astfel de cazuri calculatorul sistează operaţia în curs de desfăşurare şi trece la execuţia altor operaţii impuse de situaţia nou creată, după ce în prealabil a analizat importanţa, oportunitatea şi priorităţile tuturor acţiunilor în aşteptare şi a ales-o pe aceea care, conform criteriilor stabilite de proiectant, are cea mai mare prioritate.

În mod uzual la calculatoarele moderne aceste priorităţi sînt diferenţiate pe 8÷24 nivele, după importanţa lor specifică în cadrul procesului tehnologic condus. Astfel, de exemplu, dacă în timp ce calculatorul efectuează operaţia de supraveghere (prioritate 8) se anunţă simultan depăşirea temperaturii admisibile într-un reactor (prioritate 2) şi scăderea sub limită a activităţii catalizatorului (prioritate 6), atunci calculatorul sistează operaţia de supraveghere şi începe procedura impusă de depăşirea temperaturii care are o prioritate mai mare decît cealaltă, necesitată de scăderea activităţii catalizatorului.

În ceea ce priveşte comanda unităţii centrale, aceasta trebuie să fie de tipul macro-programabilă, fiind necesar să asigure:

— flexibilitatea setului de instrucţiuni necesară optimizării diverselor procese;

— compatibilitatea mai multor tipuri de echipamente în cadrul instalaţiei de calcul sau a reţelei de calculatoare;

— adaptabilitatea pe mai multe calculatoare a unor pachete de programe pentru cele mai uzuale aplicaţii;

— sporirea vitezei de executare a celor mai uzuale operaţii şi subrutine.

Unităţile de memorie auxiliară. În afară de memoria de lucru a *UC* care este folosită pentru memorarea programelor în execuţie, a datelor şi rezultatelor curente, calculatorul de proces dispune şi de cîteva unităţi

Scanned with OKEN Scanner

de memorie auxiliară menite să lărgească posibilităţile sale, pentru a face faţă unor aplicaţii mai ample.

Pentru memorarea unor date şi programe la care se apelează mai rar şi care nu reclamă o mare urgenţă în manipulare, se folosesc unităţi de memorie auxiliară de mare capacitate cu discuri, cu tamburi sau cu benzi magnetice.

Faţă de unităţile de memorare cu miezuri magnetice ale *UC*, care sînt mult mai scumpe şi vulnerabile la pierderea informaţiei prin citire sau prin pană de alimentare cu energie electrică, unităţile de memorare auxiliare (mai ales cele cu discuri magnetice) prezintă serioase avantaje, care le asigură în afară de mari capacităţi de memorare, de ordinul milioanelor de cuvinte, şi un timp de acces rezonabil.

Unităţile de interfaţă calculator-operator. Calculatorul de proces oferă mai multe posibilităţi de comunicare cu operatorii decît calculatoarele universale. Comunicarea are loc între calculator pe de o parte şi personalul (operatorii) de exploatare a calculatorului şi personalul (operatorii) de conducere a instalaţiei tehnologice automatizate pe de altă parte.

Echipamentul (unităţile) de comunicare cu personalul de exploatare a calculatorului este practic identic cu echipamentul calculatoarelor universale; este vorba de:

— *consola operatorului*, ca unitate de comunicare bilaterală cu *UC* prin diverse dispozitive de intrare/ieşire;

— unul sau mai multe *cititoare* de cartelă sau de benzi, ca unităţi de intrare;

— una sau mai multe *imprimante*, ca unităţi de ieşire;

— unul sau mai multe *perforatoare* de cartele sau benzi, ca unităţi de ieşire ş.a.

Aceste echipamente sînt amplasate de obicei în acelaşi loc (încăpere) cu unitatea centrală şi exploatate de personal specializat.

Echipamentul de comunicare cu personalul de conducere a instalaţiei tehnologice are un caracter oarecum specific procesului tehnologic condus. Comunicarea se face prin forme şi mijloace tehnice variate într-un limbaj simplu şi comun unei mari categorii de oameni: operatori tehnologi, mecanici, electricieni etc.

Din acest gen de echipament merită a fi menţionate diverse tipuri de:

— *dispozitive de introducere manuală* prin butoane, chei, manete ş.a. a diverselor comenzi asupra elementelor de execuţie, asupra regulatoarelor, dispozitivelor de semnalizare, asupra dispozitivelor de comutare de pe „manual" pe „automat" şi invers ş.a.;

— *dispozitive de vizualizare a informaţiei* prin cadrane, scale, tuburi catodice de tip „display" etc.;

— *dispozitive de semnalizare optică* şi/sau acustică (prin lămpi colorate, sirene ş.a.);

— *maşini rapide* pentru transmiterea mesajelor scrise şi înregistrarea evenimentelor importante (depăşiri de limite, opriri ş.a.);

— *maşini speciale* de imprimat pe coloane;

— *dispozitive de înregistrat* evoluţia în timp de tipul „digital plotter"-elor etc.

Scanned with OKEN Scanner

Aceste echipamente cu elementele lor finale se amplasează în camera dispecerului de așa manieră încît să sugereze natura și locul de proveniență a informației sau comenzii transmise.

Echipamentul de interfață proces-calculator

Echipamentul de interfață proces-calculator este specific calculatoarelor de proces și merită o analiză mai detaliată.

Așa cum reiese din figura 10.13 calculatoarele de proces sînt echipate cu mai multe feluri de unități de intrare/ieșire care mijlocesc o mulțime de canale de legătură cu procesul tehnologic și cu personalul de conducere. Tipul și numărul acestor canale este determinat de natura și complexitatea procesului tehnologic condus, de natura și volumul de informație transmisă, precum și de natura și numărul comenzilor ce trebuie efectuate de calculator.

Canalele de intrare/ieșire (I/E). Au o serie de proprietăți comune ce decurg din caracterul asincron de funcționare a acestora în raport cu UC, din diferența de viteză de lucru a acestor canale și UC, precum și din necesitatea de a asigura un transfer suplu de informație fără timpi morți sau supraîncărcări.

De regulă, unitățile și canalele de I/E au viteze de lucru mai mici decît UC, ceea ce ar limita viteza de lucru a calculatorului dacă toate canalele de comunicație cu calculatorul ar fi conectate direct cu UC. Pentru a diminua sau elimina acest impediment se adoptă toate cele trei forme de introducere și transfer de date pe baza unei programări corespunzătoare a calculatorului.

De asemenea, pentru a descărca UC de anumite funcții simple cu caracter repetativ pentru fiecare canal, constructorul de calculatoare recurge la înglobarea unor funcții de memorare, de calcul și de comandă în însăși canalele sau unitățile de I/E (de pildă : compararea unei variabile măsurate cu o valoare de referință pentru a semnaliza o eventuală depășire, însumarea sau medierea mai multor valori ș.a.).

Tot pentru mărirea supleței, unitățile de I/E moderne sînt dotate cu registre tampon proprii care păstrează provizoriu date și/sau instrucțiuni de exploatare.

Canalele de comunicație cu procesul trebuie să ofere calculatorului posibilitatea de a se adapta ușor și economic la o mare varietate de situații, adică să dispună de cîteva zeci sau sute de intrări analogice și numerice, de cîteva zeci de asemenea ieșiri, de cîteva dispozitive de ținere a evidenței și de afișare a informației ș.a., cu posibilități de adăugare sau suprimare.

În continuare ne propunem o succintă descriere a canalelor de legătură cu procesul.

Canale de intrare analogică. Un mare volum de informație privind desfășurarea proceselor tehnologice este furnizat de traductoare de tip analogic și transmise prin canale adecvate sub această formă. Funcțiile principale ale acestor canale sînt :

— colectarea și centralizarea informației analogice provenind de la proces ;

Scanned with OKEN Scanner

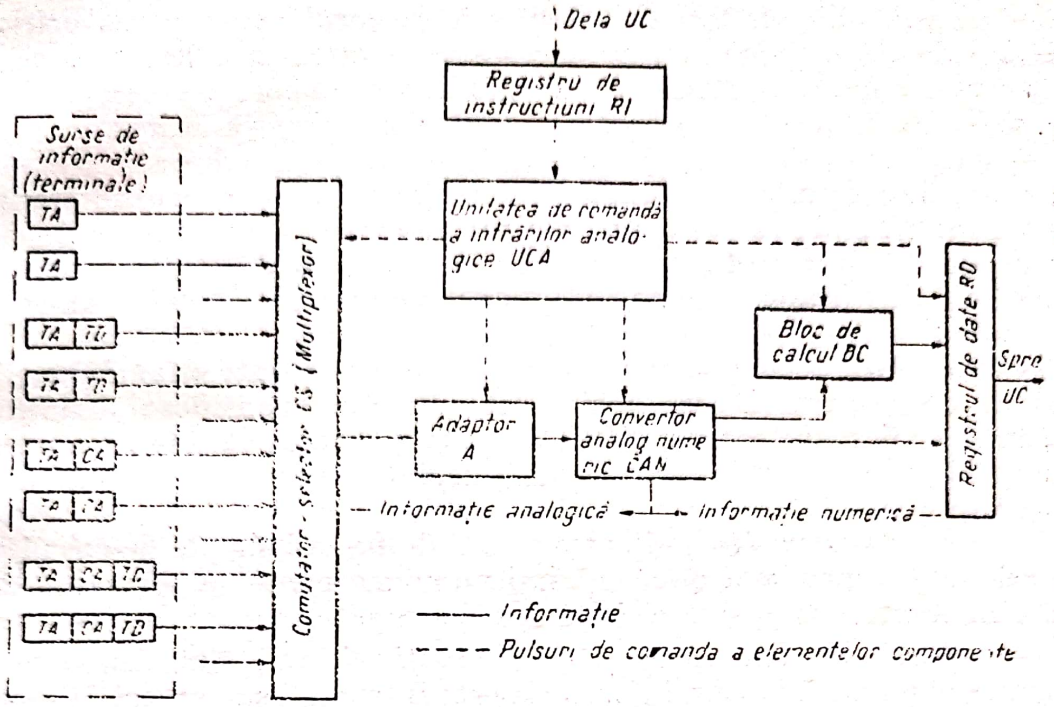


Fig. 10.14.

— transformarea și standardizarea semnalelor purtătoare de informație pentru a putea fi recepționate corect de către un amplificator sau convertor unic ;

— selectarea liniei de intrare pentru recepția informației și transpunerea acesteia într-un echivalent (cod) numeric specific calculatorului ;

— eventuala prelucrare a informației recepționate : comparări și semnalizări, însumări, etalonări, corecții ș.a. ;

— transferarea informației astfel tratate spre *UC* sau spre diverse echipamente periferice ale calculatorului ș.a.

În figura 10.14 este prezentat modelul simplificat al ansamblului de circuite (linii) de intrare analogice și fluxul de informație care circulă de la proces la calculator printr-un canal analogic multiplu, ce poate conține zeci sau chiar sute de linii analogice simple.

Din cadrul acestui ansamblu remarcăm următoarele elemente componente și funcțiile lor :

— *traductoarele analogice* (*TA*) care transpun valorile unei variabile a procesului pe un semnal purtător de informație de tip analogic, fiind deci surse de informație.

Dacă informația este transpusă pe un semnal electric ea poate fi transmisă direct la calculator ;

— *dispozitive de transmitere de la distanță* (*TD*) care mijlocesc transmisia către calculator a semnalelor de ieșire ale traductoarelor electrice în cazul cînd distanța dintre aceste elemente este mare, iar traductorul nu poate asigura o transmitere fără distorsiuni a informației ;

— *convertoarele analogice* (*CA*) care transpun informație furnizată de traductoare de tip neelectric pe un semnal purtător de informație de tip electric (tensiune sau curent), ce se pretează mai bine transmiterii la distanță ;

Scanned with OKEN Scanner

— *comutatorul selector* (*CS*) sau *multiplexorul* este un dispozitiv de comutare de tip matrice care mijlocește conectarea la calculator, pe rînd sau la cerere, a fiecăreia din sursele de informație așa cum prevede programul de exploatare. El are deci numeroase intrări și o singură ieșire ;

— *adaptorul analogic* (*A*) care are funcția de încadrare a semnalelor purtătoare de informație între limitele admisibile ale convertorului analog-numeric, prin amplificarea, atenuarea sau divizarea acestor semnale, astfel încît să devină standardizate. El funcționează prin divizarea timpului în fiecare linie de intrare și poate îndeplini și funcția de filtrare a semnalelor de intrare ;

— *convertorul analog-numeric* (*CAN*) care convertește semnalul analogic standard de intrare într-un echivalent numeric cuantificat (eșantionat) compatibil cu limbajul intern al *UC*. Are intrare analogică și ieșire numerică, iar unele tipuri (cele cu integrare) pot îndeplini și funcția de filtrare a semnalelor ;

— *blocul de calcul* (*BC*) eliberează *UC* de funcții simple de calcul, cum ar fi cele de comparare a două mărimi numerice pentru a semnaliza depășirea unor limite, însumarea a două sau mai multe mărimi, media lor ș.a. ;

— *registrele de date* (*RD*) și *de instrucțiuni* (*RI*) sînt dispozitive de memorare temporară în care informația este introdusă sau scoasă la momente de timp determinate de o funcționare optimă a *UC*, constituind un acumulator tampon de informație ;

— *unitatea de comandă* (*UCA*) care asigură coordonarea acțiunilor tuturor elementelor componente ale ansamblului pe baza instrucțiunilor primite de la *UC*. Această unitate decodifică instrucțiunile primite de la *UC*, alege sursa de informație pentru măsurare prin intermediul comutatorului selector, generează semnale de temporizare a acțiunilor, inițiază și efectuează conversia analog numerică și eventual anumite funcții de calcul, stabilește deci întregul regim de funcționare a ansamblului, inclusiv întreruperea unui ciclu început, dacă *UC* cere execuția altui ciclu cu prioritate superioară.

Evident pot exista și alte variante de construcție și funcționare a unor astfel de echipamente cu mai puține sau mai multe din funcțiile prezentate. Aici s-au prezentat cele mai caracteristice aspecte legate de subiectul abordat.

Canale de intrare numerice și logice. Un volum apreciabil de informație privind variabilele de stare poate fi furnizat de *traductoarele de tip binar* (logic) *TB* care indică starea respectivă prin una din cele două valori binare : da sau nu (adevărat sau fals). Astfel de traductoare pot indica poziția ventilelor și contactelor de tip închis/deschis, depășiri de limite ș.a.

De asemenea, valorile unor variabile pot fi exprimate prin frecvența pulsurilor generate de *traductoare numerice* de tip electric, cum ar fi tahogeneratoarele cu pulsuri, debitmetre-generatoare de pulsuri ș.a.

Introducerea în calculator a informației transpuse pe un astfel de semnal discret se face fără convertirea în alt cod numeric.

Un canal multiplu pentru astfel de semnale îndeplinește următoarele funcții :

— colectarea și centralizarea informației de acest gen ;

— selectarea liniilor de intrare și adaptarea pulsurilor recepționate la caracteristicile de intrare ale elementului receptor ;

Scanned with OKEN Scanner

— transferul informației astfel obținute către *UC*;

— determinarea frecvenței pulsurilor pentru a transpune informația recepționată în codul intern al calculatorului ș.a.

În figura 10.15 este prezentată schema simplificată a unui ansamblu de linii de intrare pentru informația de tip logic și numeric din care reiese și fluxul de informație de la proces la calculator. Dintre elementele componente și funcțiile lor se remarcă:

— *traductoarele binare* (*TB*) care au același rol ca și traductoarele analogice cu deosebirea că transpun informația pe semnale binare;

— *traductoarele numerice* (*TN*) care transpun informația privind variabilele procesului condus pe un tren de pulsuri modulat, de obicei în frecvență sau fază (poziție);

— *comutatorul selector* (*CS*) sau *multiplexorul* care, ca și în cadrul canalelor analogice, mijlocește selectarea liniei de intrare după regulile stabilite de *UC* pentru a prelua prin eșantionare informația transmisă de fiecare sursă de informație;

— *adaptorul de pulsuri* (*AP*) care adaptează semnalele binare și pulsurile recepționate la caracteristicile de lucru ale calculatorului: amplitudine, frecvență ș.a.m.d.;

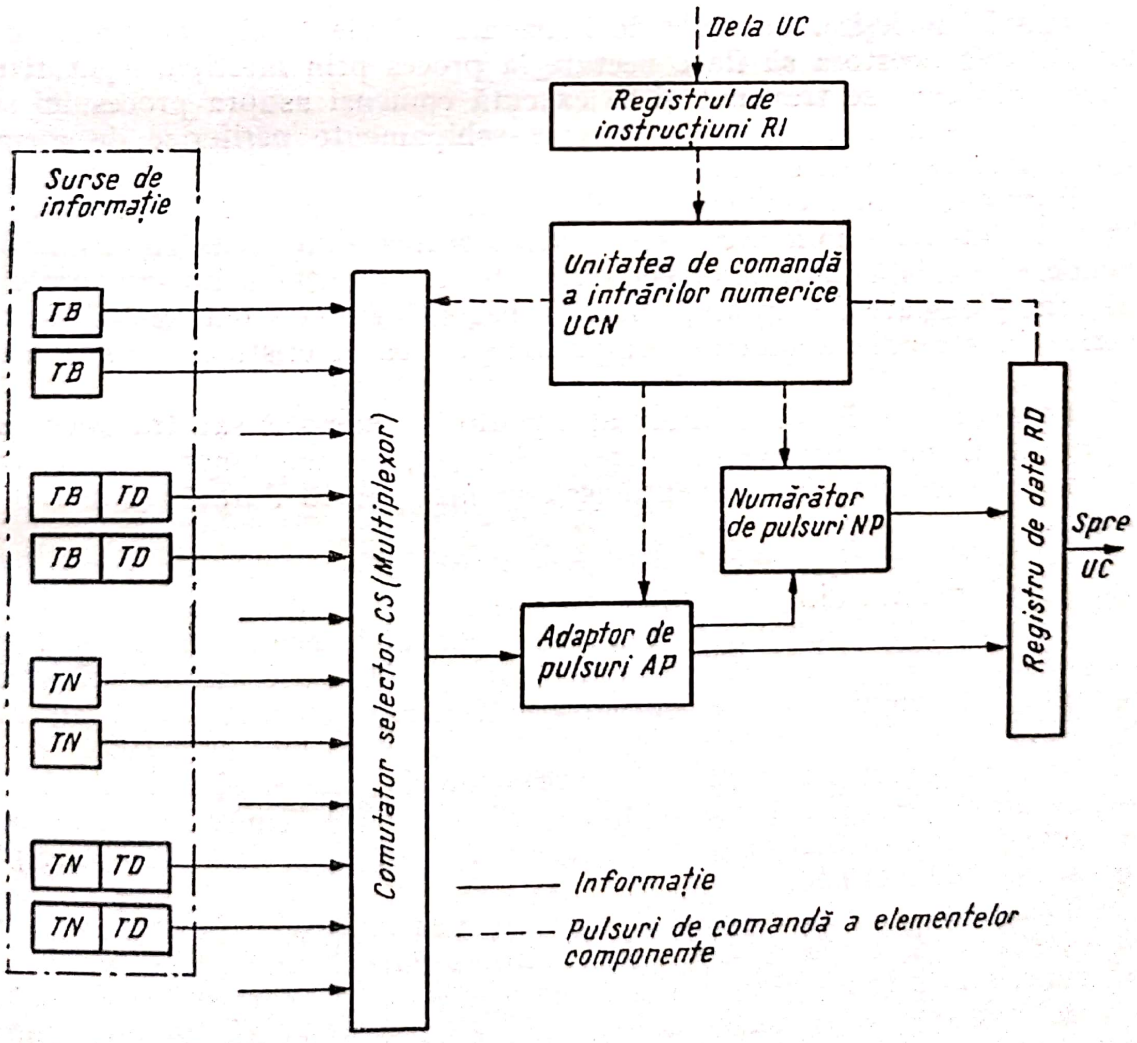


Fig. 10.15.

— *numărătorul de pulsuri* (*NP*) care determină pe baza numărării pulsurilor recepționate într-un interval de timp standard (durata de eșantionare) echivalentul numeric al variabilei transpuse pe trenul de pulsuri;

— *registrele de instrucțiuni* (*RI*) și *de date* (*RD*) care au funcția de memorare de scurtă durată a instrucțiunilor date de *UC* și a datelor obținute, pînă la anumite momente de timp determinate de funcționarea asincronă a elementelor componente ale calculatorului;

— *unitatea de comandă* (*UCN*) care îndeplinește, ca și *UCA*, funcția de coordonare a acțiunilor tuturor elementelor acestui ansamblu pe baza instrucțiunilor primite de la *UC*.

La descrierea acestor echipamente s-au avut în vedere aspectele cele mai generale și mai caracteristice. Diversele echipamente realizate de diverse firme pot prezenta, desigur, anumite diferențieri. Așa de pildă, constructorii de calculatoare pot moduliza canalele analogice și pe cele numerice pentru un număr oarecare de variabile, de ordinul zecilor sau sutelor, repetînd elementele unicat (comutator, convertor, adaptor, registre etc.) pentru fiecare modul, sau pot reuni într-un singur modul (bloc) și canalele analogice și pe cele numerice.

Canale de ieșire. Canalele de ieșire ale calculatoarelor de proces dau posibilitatea acestora să fie conectate la proces prin multiple legături inverse, prin care se transmit și se execută comenzi asupra procesului sau pentru conectarea acestora la diverse echipamente periferice de memorare, vizualizare sau semnalizare.

Forma cea mai convenabilă de obținere a informației și comenzilor din calculatoarele numerice este evident forma numerică. În instalațiile tehnologice există însă, de obicei, elemente de execuție și unele aparate de măsurat și reglare de tip analogic care necesită convertirea semnalelor numerice în semnale analogice compatibile cu caracteristicile acestor echipamente.

Prin canalele de ieșire informația poate fi transmisă sau transpusă pe: semnal analogic (tensiune sau curent electric), semnal binar electric, tren de pulsuri modulate în frecvență (sau în fază sau în durată) și în cod de pulsuri.

Canalele și echipamentele de ieșire pentru semnal analogic sînt relativ scumpe și prezintă dificultăți de construcție. Ele constituie o varietate mai puțin uzuală la calculatoarele numerice. Semnalele analogice se obțin prin comanda unei rețele de rezistențe de precizie conectate astfel încît să dea la ieșire o tensiune sau un curent electric corespunzător comenzii date de *UC*.

Cele mai multe ieșiri ale calculatoarelor de proces sînt de tipul tren sau coduri de pulsuri chiar și pentru ieșirile analogice. Un exemplu de canal de ieșire pentru semnal analogic provenit dintr-un tren de pulsuri este prezentat în figura 10.16.

Trenul de pulsuri acționează motorul pas cu pas în poziția necesară, iar acesta modifică în mod corespunzător semnalul de intrare în convertor pentru a da la ieșire un semnal analogic standard care stabilește valoarea de referință a regulatorului și/sau valoarea indicată de aparatul de vizualizare corespunzător.

Scanned with OKEN Scanner

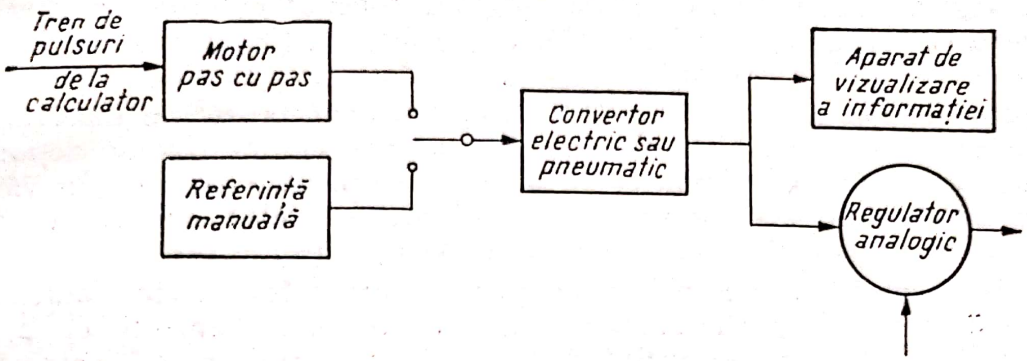


Fig. 10.16.

Canalele și ieșirile numerice sub formă de tren de pulsuri sînt destinate în principal pentru comanda diferitelor dispozitive cum ar fi: circuitele basculante, redresoarele comandate, releele, electromagneții, starterele motoarelor, dispozitivele de afișare, perforatoarele de benzi, imprimantele ș.a.

În figura 10.17 este prezentată o configurație posibilă a canalelor de ieșire ale unui calculator de proces.

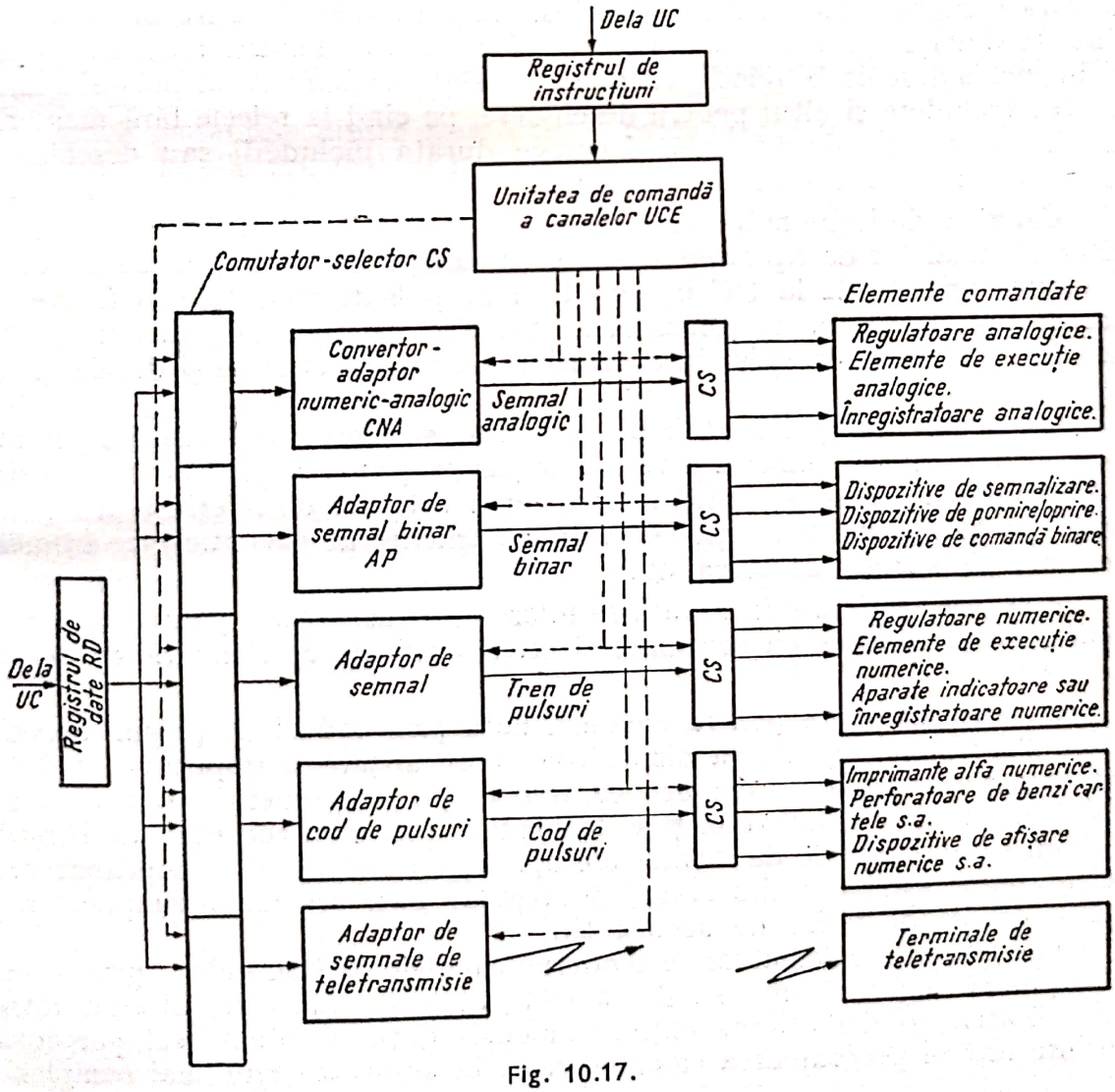


Fig. 10.17.

Scanned with OKEN Scanner

Instrucțiunile și datele de la *UC* sînt transmise prin intermediul *registrelor* (*RI* și, respectiv, *RD*) *unității de comandă* (*UCE*) a canalelor de ieșire, care le preia și le execută.

Comutatoarele selectoare (*CS*) asigură conectarea calculatorului cu dispozitivul sau dispozitivele de ieșire asupra cărora trebuie acționat la anumite intervale de timp și pe durate determinate de *UC*. În varianta prezentată, selecția și comutarea se fac la două nivele: la nivelul tipului de semnal și la nivelul liniei de ieșire de același tip.

Elementele ce urmează după primul comutator-selector, adică: convertoare-adaptoare, schimbătoare de cod etc., sînt elemente cu ajutorul cărora se elaborează semnalul de ieșire în conformitate cu instrucțiunile și datele primite de la *UC*, în așa fel încît să fie compatibil cu caracteristicile de intrare ale elementelor comandate: regulatoare, înregistratoare, perforatoare de cartele ș.a. Întrucît aceste elemente comandate primesc informația sau comenzile în mod discret, ele trebuie să aibă posibilitatea (mijloacele necesare) de memorare pe durata unui ciclu a comenzii sau informației transmise de *UC*.

Cel mai simplu canal de ieșire numerică este cel de tip releu care asigură comenzi binare: închis/deschis. Dispozitivele de ieșire de acest tip sînt de două feluri: cu memorie și fără memorie. Pentru realizarea unui ciclu închis/deschis la releele cu memorie sînt necesare două pulsuri: unul pentru închidere și altul pentru deschidere, pe cînd la releele fără memorie este suficient un singur puls, deoarece durata închiderii sau deschiderii coincide cu durata pulsului respectiv.

Canalele de ieșire sub formă de tren de pulsuri constituie un caz particular al canalelor de tip releu asociate cu funcția de transformare a valorii numerice primite de la *UC* într-un tren de pulsuri modulate în frecvență sau durată. Unitatea de comandă a canalelor de ieșire reglează frecvența sau durata pulsurilor emise în concordanță cu instrucțiunile și datele emise de *UC*.

Unele din principalele utilizări ale ieșirilor sub formă de pulsuri modulate în frecvență este acționarea motoarelor pas cu pas, a registrelor cu circuite basculante ș.a., folosite pentru introducerea valorilor de referință la regulatoare, precum și pentru acționarea unor elemente de execuție care primesc un astfel de semnal de comandă.

Pulsurile modulate în durată se folosesc, de asemenea, pentru comanda unor servomecanisme cu acționare electrică utilizate ca elemente de execuție ș.a.

Canalele de ieșire pentru comenzi date prin coduri de pulsuri servesc îndeosebi pentru acționarea dispozitivelor de afișare și imprimare a datelor. Prin aceste canale și dispozitive, informația sau comenzile se preiau din codul intern al calculatorului și se transpun pe un nou cod numeric compatibil cu caracteristicile de intrare ale aparatelor comandate: imprimatoare, perforatoare ș.a. care deservesc, de regulă, prin divizarea timpului mai multe canale de ieșire de același tip.

La unele din calculatoarele moderne canalele de ieșire către proces sînt de tipul cu legătură inversă de la elementele de execuție, oferind astfel posibilitatea confirmării execuției comenzii date. Evident, echipamentul de interfață și programarea calculatorului în acest caz sînt mai complexe.

Scanned with OKEN Scanner

Canale şi echipamente terminale. Un alt mod caracteristic de comunicare sau telecomunicare cu calculatorul de proces îl constituie canalele de intrare/ieşire legate la aşa-zisele terminale.

Terminalele sînt echipamente periferice de intrare şi/sau ieşire instalate la utilizatorii calculatorului în apropierea surselor de informaţie sau în locurile unde se recepţionează rezultatele prelucrării informaţiei sau comenzile (deciziile) date de calculator. Prin ele se transmit către calculator date şi eventual programe şi/sau se difuzează rezultatele prelucrării informaţiei sau orice informaţie provenind de la calculatorul cu care sînt conectate.

Într-o accepţiune mai largă orice element primar sau final al unui canal de transmitere poate fi numit un terminal. În literatura de specialitate s-a consacrat această denumire numai pentru elementele care realizează comunicarea prin intermediul omului.

Ca terminale pentru canalele de introducere a informaţiei în calculator se pot folosi : maşini de scris cu claviatură, tuburi catodice speciale, cititoare de cartele sau benzi perforate, cititoare de benzi magnetice, diverse traductoare conectate la proces, calculatoare terminale ş.a.

Ca terminale pentru canalele de recepţie a informaţiei sau a comenzilor se pot folosi : maşini de scris sau imprimante rapide, tuburi catodice speciale, perforatoare de cartele sau benzi, înregistratoare magnetice, panouri cu reţea de lămpi, diverse elemente de execuţie legate la proces, calculatoare ş.a.

Dacă distanţele dintre terminale şi calculatorul principal sînt relativ mici (pînă la 600 m), legăturile dintre terminale şi calculator se realizează prin linii individuale. Dacă însă numărul terminalelor şi distanţele pînă la calculator sînt relativ mari, sînt necesare dispozitive de adaptare la liniile de transmisie şi este mult mai avantajos ca legătura cu calculatorul să se facă printr-un canal comun la capetele căruia să se monteze cîte un multiplexor sau un concentrator de linii ca în figura 10.18. Diversele terminale T sînt conectate la un multiplexor M sau la un concentrator de linii CL, prin linii individuale relativ scurte (LS).

Întrucît în calculator informaţia este transpusă în cod de pulsuri binare, este necesar ca şi informaţia emisă sau recepţionată de terminale să fie transpusă tot în acest cod de pulsuri. În acest scop terminalele, care în mod normal emit sau recepţionează informaţia în alt cod decît codul binar, sînt cuplate cu convertoare-adaptoare (CA) corespunzătoare. Multi-

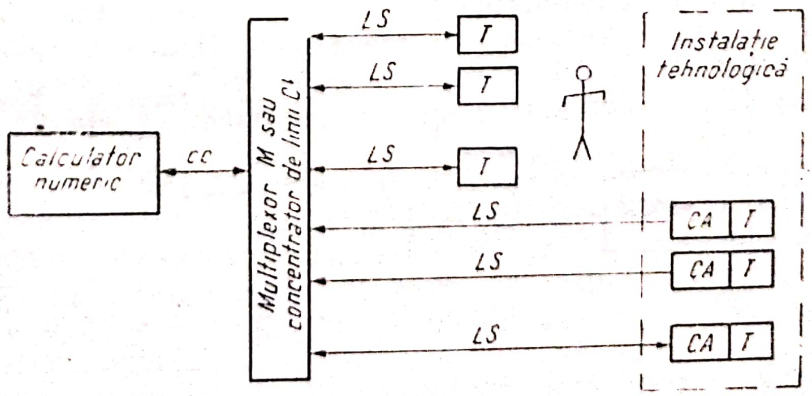


Fig. 10.18.

plexorul selectează cîte unul din numeroasele terminale ca să-l conecteze la canalul comun *CC*, pentru a-l pune în legătură cu calculatorul, sau distribuie informaţia primită prin acest canal succesiv la fiecare terminal. Concentratorul de linii *CL*, care poate fi folosit în locul multiplexorului, este înzestrat cu un dispozitiv de memorare tampon în care se colectează informaţia de la numeroasele terminale conectate prin linii de transmitere individuale de mică viteză, ca să o transmită prin canalul comun cu viteză mult mai mare, sau invers (distribuie informaţia primită prin canalul rapid spre numeroasele terminale succesiv, prin linii de transmitere mai lente).

Ca linii de transmitere pot fi folosite reţelele telegrafice, liniile telex, liniile telefonice ş.a., iar sensul de transmitere poate fi unic, de la terminale la calculator (cazul centralizării datelor), de la calculator la terminale (cazul difuzării informaţiei), sau poate fi bilateral. În acest din urmă caz, transmisia poate fi alternativă, pe o singură linie, de genul întrebare într-un sens şi răspuns în celălalt sens sau poate fi simultană, dar în acest caz sînt necesare două linii: una pentru un sens, iar cealaltă pentru celălalt sens.

Dacă distanţa dintre terminale şi calculator este mare (de ordinul zecilor de kilometri), efectul perturbaţiilor asupra transmisiei informaţiei este atît de mare încît o face aproape imposibilă. Din această cauză, se recurge la transpunerea informaţiei pe un semnal purtător de informaţie, insensibil la perturbaţii, prin procedeul de *mo*dulare-*dem*odulare realizat de dispozitive adecvate denumite pe scurt *MODEM*.

Cel mai frecvent se recurge la modularea în frecvenţă sau fază a unui semnal sinusoidal de către un semnal binar purtător al informaţiei de transmis, prin intermediul unor *MODEM*-uri pentru ambele sensuri de transmitere, plasate aşa ca în figura 10.19. Schema din figura 10.19 nu se deosebeşte de cea din figura 10.18 decît prin aceea că în canalul comun de transmitere din figura 10.19 este intercalat un *MODEM* la intrarea/ieşirea calculatorului şi un alt *MODEM* la intrarea/ieşirea multiplexorului sau concentratorului de linii, situat în zona utilizatorilor terminalelor. În acest fel canalul de transmitere devine practic insensibil la perturbaţii, fiind şi economic (pentru că este comun tuturor utilizatorilor).

Comunicarea şi telecomunicarea prin terminale asigură calculatoarelor de proces o serie de noi utilizări şi avantaje, dintre care menţionăm:

— deservirea mai multor utilizatori care nu dispun de calculatoare proprii dar care doresc să beneficieze de serviciile unui calculator central în condiţii economice convenabile;

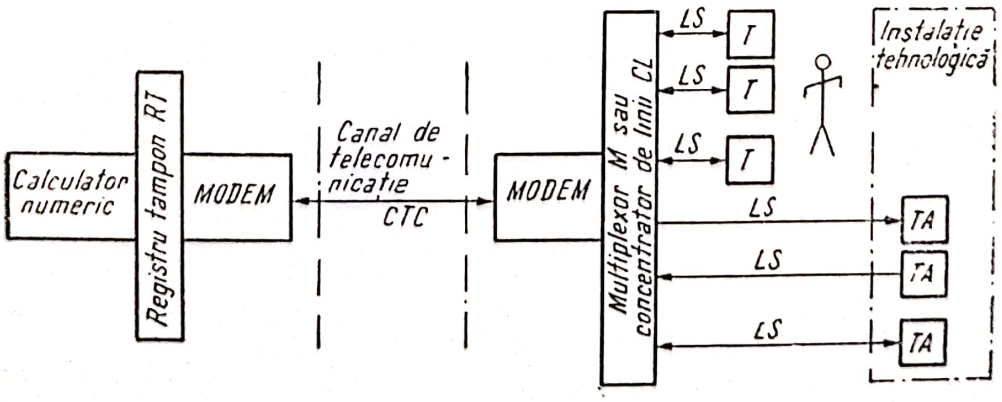


Fig. 10.19.

— utilizarea mai raţională (mai eficientă) a calculatoarelor de către mai mulţi utilizatori ;

— posibilitatea utilizării în timp real a unei reţele de calculatoare şi terminale pentru conducerea marilor complexe industriale, economice sau sociale în aplicaţii tehnice ca : folosirea în comun de către mai mulţi utilizatori a informaţiei concentrate în mari fişiere, baze sau bănci de date ; colectarea informaţiei de pe mari întinderi geografice în vederea folosirii ei de către mai mulţi utilizatori ; difuzarea informaţiei la numeroşi abonaţi ; conducerea unor activităţi care se desfăşoară dispersat ş.a.

Toate acestea explică evoluţia atît de rapidă a utilizării reţelelor de calculatoare în cele mai variate domenii de activitate, dintre care, bineînţeles, şi controlul şi conducerea proceselor tehnologice. Există în lume deja numeroase asemenea reţele de calculatoare de proces şi terminale aranjate astfel încît să poată conduce în colectiv mai multe procese situate la distanţe mari, pe baza unei subordonări în repartizarea funcţiilor, ţinîndu-se cont de disponibilităţi şi a unui ajutor reciproc în caz de supraîncărcare sau defectare a vreunuia din calculatoare sau terminale.

În figura 10.20 este prezentată o schemă simplificată a unei reţele de calculatoare pentru o centrală industrială, care coordonează activitatea a n întreprinderi. În schemă sînt utilizate următoarele notaţii : CPC — calculator principal de comandă şi comutare a mesajelor ; $CC1, CC2, \ldots, CCn$ — calculatoare de conducere ; $CPij$ $(i, j = 1, 2, \ldots, n)$ — calculatoare de proces ; UPI — unităţi periferice de intrare locale ; UPE — unităţi periferice de ieşire locale ; T — terminale ; LT — linii de telecomunicaţii ; PO — programe de optimizare ; S — senzori şi E — elemente de execuţie pentru modificarea parametrilor proceselor. În cadrul unei astfel de reţele minicalculatoarele de concepţie modulară pot avea o pondere importantă datorită facilităţilor acestora de a forma structuri optime pentru o mare varietate de aplicaţii.

Alegerea componentelor calculatoarelor de proces

Calculatoarele de proces sînt de fapt instalaţii complexe de colectare, centralizare şi prelucrare a informaţiei şi eventual de comandă nemijlocită, compuse din elemente componente modulare care permit obţinerea unei game largi de configuraţii. În aceste condiţii, se pune problema alegerii acelei configuraţii care asigură cea mai înaltă eficienţă tehnico-economică a procesului condus.

Pentru a face o astfel de alegere este necesar să se determine natura şi volumul informaţiei care trebuie centralizată şi prelucrată, natura şi volumul comenzilor de executat, precum şi tehnicile de realizare a acestor activităţi.

Alegerea configuraţiei optime a unui calculator de proces se face, în principal, după criteriul concordanţei dintre necesitate şi resurse.

Funcţiile şi performanţele funcţionale ale unui calculator de proces trebuie apreciate după modul în care corespund cerinţelor procesului tehnologic condus şi obiectivelor urmărite prin folosirea sistemului de calcul.

Performanţele funcţionale cum ar fi : viteza de lucru a unităţii centrale, timpul de acces la memorie, viteza de transfer de informaţie ş.a. trebuie

Scanned with OKEN Scanner

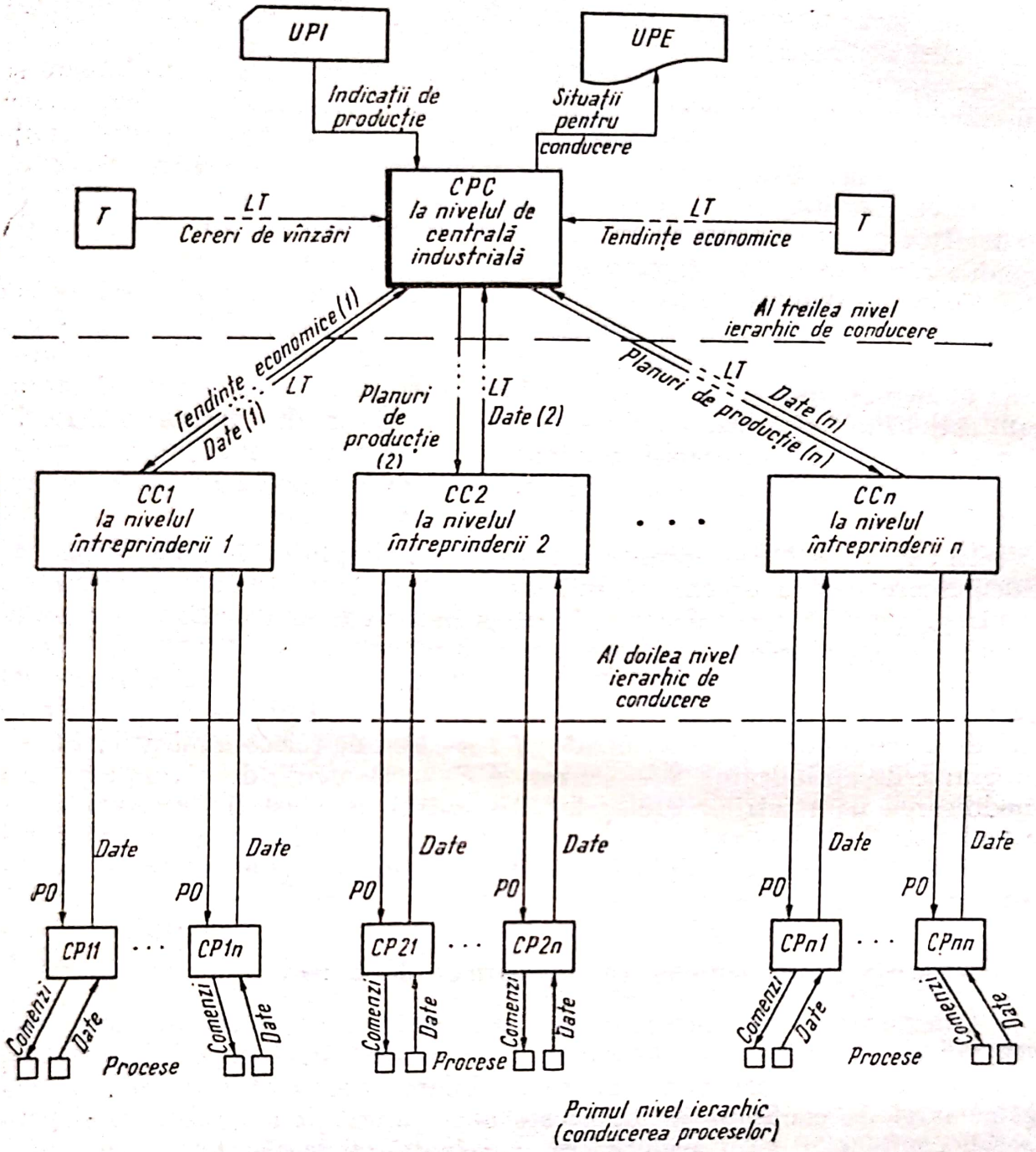


Fig. 10.20.

apreciate nu numai după valorile lor absolute ci şi după modul în care se armonizează între ele. De exemplu, o unitate centrală cu viteză de prelucrare foarte mare ar putea fi nepotrivită pentru o instalaţie de calcul cu capacitate de memorare insuficientă, cu canale de intrare/ieşire prea lente sau care nu poate îndeplini anumite funcţii specifice procesului condus.

Trebuie, deci, să se pună de acord viteza de colectare și centralizare a informaţiei cu viteza de prelucrare, cu capacitatea de memorare şi cu capacitatea de transfer a acesteia, precum și cu capacitatea de a emite și executa comenzi în cazul conectării calculatorului la proces.

Scanned with OKEN Scanner

De asemenea, trebuie să se aibă în vedere posibilitatea cuplării fiecărui element component al unei instalaţii de calcul cu alte instalaţii, pe principiul colaborării sau subordonării, ceea ce este pe deplin posibil dacă se utilizează componente modulare standardizate.

10.3. INTRODUCERE ÎN PRACTICA MODELĂRII ŞI IDENTIFICĂRII PROCESELOR TEHNOLOGICE

Cu ajutorul calculatoarelor automate se poate rezolva practic orice problemă care poate fi algoritmizată şi programată. Pentru aceasta este însă necesar să definim cu claritate, de la început, problema avută în vedere prin: scopul, datele de plecare, modalitatea de soluţionare pas cu pas, forma de prezentare a soluţiei [11] ş.a. În acest scop este necesar să se efectueze o atentă identificare a proceselor şi o corectă modelare a acestora.

Probleme generale ale modelării

Prin *model* al unui sistem se înţelege o reprezentare simbolică, logică şi/sau cantitativă, a unor relaţii esenţiale existente în cadrul sistemului, eventual prin neglijarea aspectelor neesenţiale în raport cu scopul urmărit de utilizator.

Elaborarea unui model de proces se face pe baza unei analize multilaterale şi temeinice a procesului pentru a identifica şi reliefa aspectele esenţiale legate de scopul urmărit, implicaţiile şi dependenţele existente între variabilele procesului, influenţele la care acesta este supus din partea unor factori interni şi externi, posibilităţile de dirijare şi de optimizare etc.

Exprimarea sau vizualizarea unui model se poate face prin unul sau prin mai multe din următoarele mijloace : descriere prin cuvinte, schema bloc, scheme logice, tabele de decizie, tabele, curbe, nomograme, funcţii, ecuaţii matematice ş.a.

Elaborarea unui model constă, aşadar, în formularea unui model abstract ce urmează a fi verificat şi perfecţionat pe baza „confruntării" sale cu originalul.

Deosebit de importantă este elaborarea corectă a modelelor de proces care se exprimă în limbaj matematic — funcţii sau ecuaţii — în vederea optimizării procesului. Pe baza unui astfel de model se stabileşte algoritmul şi programul de rezolvare (simulare) pe calculator a problemei modelate pentru găsirea soluţiei, care este preluată de factorii de decizie şi folosită pentru conducerea procesului. Aşadar, în acest caz baza o constituie modelul, mijlocul este simularea, iar scopul este optimizarea.

Elaborarea unui model comportă următoarele *obiective* şi *etape* :

— definirea scopului căruia să-i servească modelul ;
— delimitarea sistemului care trebuie modelat în acest scop ;
— precizarea variabilelor şi restricţiilor semnificative ;
— alegerea tipului de model potrivit cu scopul urmărit ;
— formularea modelului ;
— simularea procesului şi eventual perfecţionarea modelului.

Scanned with OKEN Scanner

Prin urmare, cu calculatorul se rezolvă o problemă abstractă reprezentată printr-un model care poate exprima toate aspectele unei probleme concrete sau numai o parte din ele, putîndu-le exprima corect sau fals; ca urmare, soluţiile obţinute pe model sînt veridice numai în măsura în care modelul reflectă în mod corect aspectele existente în cadrul sistemului modelat. Rezultă deci că cerinţa esenţială faţă de orice model este ca acesta să fie o reprezentare cît mai fidelă a procesului modelat şi cît mai simplă.

Practica a arătat că principala piedică în conducerea proceselor, pe măsura marilor posibilităţi ale calculatoarelor, o constituie lipsa sau slaba calitate a modelelor de proces. Din această cauză identificarea procesului pentru stabilirea modelului trebuie tratată simultan şi în strînsă legătură cu sinteza sistemului de conducere pentru a se depista din timp eventualele deficienţe ale modelului şi a le înlătura pe baza analizei interacţiunii acestuia cu celelalte elemente ale sistemului.

Din nefericire, nu se poate indica un mod standard de elaborare a modelelor întrucît în afară de abordarea ştiinţifică a problemei este nevoie, în multe cazuri, să se dea cîmp liber intuiţiei creatoare. În cele ce urmează vom face unele precizări generale, cu exemplificări, legate de problematica propusă, urmînd obiectivele şi etapele enumerate mai înainte.

Elementele esenţiale ale unui model

Modelarea unui proces se face în strînsă concordanţă cu scopul căruia îi serveşte modelul. În funcţie de scop se delimitează sistemul modelat, se precizează variabilele acestuia, se alege tipul de model ş.a. Unul şi acelaşi sistem poate avea mai multe modele; de pildă, un model pentru proiectare, un model pentru exploatarea acestuia după un criteriu de performanţă tehnică, un model de exploatare după un criteriu de performanţă economică ş.a.

În cadrul diverselor modele ale aceluiaşi proces, natura şi rolul variabilelor pot fi diferite, în funcţie de scop. Aşa de pildă, cînd se proiectează un proces, anumite variabile ca: productivitatea instalaţiei, caracteristicile materiei prime, caracteristicile produselor fabricate ş.a. sînt considerate ca parametri daţi, pe cînd parametri ca: dimensiunile şi configuraţia diverselor aparate tehnologice din cadrul instalaţiei, capacitatea pompelor şi compresoarelor ş.a. sînt considerate ca variabile pe care proiectantul trebuie să le determine (să le optimizeze).

La un model de exploatare, al unui proces în desfăşurare, situaţia este alta. Mărimile care în modelul de proiectare erau variabile, aici sînt considerate ca parametri daţi, iar cele care erau fixe sînt considerate ca variabile. Reiese deci că un model formulat pentru un scop nu poate fi utilizat în alt scop.

După ce scopul modelului a fost stabilit urmează să se precizeze limitele acestuia. Pentru modelul unui proces tehnologic trebuie să se stabilească de la început care din elementele din amontele şi din avalul procesului modelat urmează a fi înglobate în model. De exemplu, modelul unui reactor poate cuprinde unităţile de separare din aval şi aparatele de schimb de căldură din amonte, sau poate fi limitat exclusiv la reactor.

Delimitînd modelul procesului se stabileşte de fapt şi natura şi caracterul variabilelor esenţiale care trebuie luate în seamă în cadrul procesului mode-

Scanned with OKEN Scanner

lat. Astfel, temperatura cu care materialul reactant intră într-un reactor poate fi considerată în mai multe feluri. Dacă încălzitorul din cadrul instalației de reacție nu este cuprins în model, acest parametru (temperatura) poate fi considerat ca variabilă independentă, pe cînd dacă face parte din model trebuie considerat ca variabilă intermediară dependentă.

Metodele și mijloacele de investigație se aleg, de asemenea, în funcție de scopul și limitele procesului modelat. Astfel, pentru studiul procesului de separare pe un taler se pleacă de la ecuațiile de conservare și de echilibru pentru talerul considerat, iar pentru măsurarea debitelor de lichid și de vapori sînt necesare metode și aparate speciale, pe cînd dacă se urmărește procesul de separare dintr-o coloană de fracționare ca un proces global se pleacă de la ecuațiile de conservare globale, iar măsurarea variabilelor ce caracterizează acest proces se poate face cu metode și mijloace obișnuite.

Variabilele și restricțiile modelelor

Sub aspect informațional variabilele unui proces se pot împărți în două categorii principale:

— variabile independente (de intrare), care reprezintă factori cauzali externi independenți de proces;

— variabile dependente (de ieșire sau intermediare), care exprimă efectele factorilor cauzali asupra procesului.

Pe de altă parte, variabilele care definesc starea sau performanțele unui proces mai pot fi denumite variabile de stare, respectiv, variabile de performanță.

Variabilele independente. Acestea pot fi necomandate sau comandate. Variabilele necomandate sînt de regulă perturbații care influențează în mod aleator desfășurarea procesului și pe care operatorul nu le poate modifica. Pentru a compensa efectul lor indezirabil se recurge la modificarea variabilelor comandate ale procesului.

Variabilele necomandate sînt generate sau legate de: materia primă sau auxiliară, mediul ambiant, starea de uzură a utilajelor sau instalației de prelucrare și de factori economici.

Variabilele ce caracterizează o materie primă pot prezenta fluctuații importante care afectează în bună măsură desfășurarea întregului proces tehnologic. De asemenea, mediul ambiant constituie în general o sursă permanentă de perturbații. În multe cazuri variabile ca: temperatura, umiditatea și presiunea mediului ambiant au o influență însemnată asupra procesului tehnologic și ar trebui incluse în model.

Uzura utilajelor constituie de asemenea o sursă importantă de perturbații. Așa de exemplu, formarea depunerilor de cocs pe tuburile cuptoarelor tubulare, depunerile de calcar sau murdărirea tuburilor schimbătoarelor de căldură, deteriorarea izolațiilor termice ș.a. sînt cîteva din cele mai importante perturbații de acest gen de care ar trebui să se țină seama la elaborarea modelului.

Factorii economici sînt prin excelență variabile aleatoare dictate de cerințele pieții, de politica de prețuri sau de alte considerente economice, de care trebuie să se țină seama. Așa de pildă, cererile sezoniere de diverse produse determină conducerile unor întreprinderi să-și reprofileze producția

Scanned with OKEN Scanner

în așa fel încît să fabrice cu precădere acele produse care se cer pe piață și care aduc astfel un beneficiu maxim.

Variabilele comandate sînt mărimi care între anumite limite pot fi modificate de către operator sub formă de comenzi cu scopul de a compensa efectele nedorite ale perturbațiilor și de a realiza conducerea efectivă a procesului.

Putem deosebi *variabile comandate primare* și *variabile comandate secundare*. Din prima categorie fac parte variabilele pe care operatorul le poate modifica nemijlocit ca, de exemplu: poziția organului de obturare al unui robinet, valoarea de referință a unui regulator, turația unei pompe ș.a. Din cea de a doua categorie fac parte o serie de variabile comandate dependente de una sau mai multe mărimi primare independente; de exemplu: în loc de a considera poziția organului de obturare al unui robinet este mai comod să se considere debitul de material care trece prin robinet, care depinde de poziția organului de obturare, de presiune ș.a.

Variabile dependente. Toate celelalte variabile ale unui proces, în afară de cele independente, sînt variabile dependente (de ieșire). Valorile lor pot fi determinate dacă se cunosc valorile variabilelor independente și relațiile dintre variabilele independente și cele dependente.

Majoritatea proceselor industriale sînt procese complexe, alcătuite dintr-un ansamblu de procese elementare care se desfășoară în serie, în paralel, cu recirculare sau mixt. În cadrul acestora au loc transformări ale substanței și/sau energiei care duc la apariția unor variabile ce definesc stări sau faze intermediare, fiind deci variabile intermediare. Valorile acestora depind de valorile variabilelor independente și determină, la rîndul lor, valorile altor variabile dependente. De exemplu, la un cuptor tubular una din variabilele de intrare este debitul de combustibil injectat în focar, una din variabilele intermediare este temperatura medie a gazelor arse în focar, iar una din variabilele dependente finale sau de ieșire este temperatura de ieșire a produsului încălzit.

În general, în cazul proceselor cu transformări succesive se poate face abstracție de variabilele intermediare, urmărindu-se dependența dintre variabilele independente și cele dependente finale, care se numesc și variabile de ieșire, prin procedeul funcție de funcție.

Variabile de stare. Starea, în desfășurarea unui proces, poate fi determinată dacă se cunosc valorile unui anumit număr minim de variabile și un număr suficient de relații între variabilele procesului.

Se poate demonstra că starea unui proces în care intervin N variabile și M relații între variabilele acestuia, poate fi pe deplin determinată dacă se cunosc valorile a

$$F = N - M \qquad (10-1)$$

variabile de stare [10].

Prin urmare, pentru definirea stării unui proces nu este nevoie să se cunoască valorile tuturor variabilelor sale, ci este suficient să se cunoască valorile a numai $N - M$ variabile. Rezultă deci că cu cît se cunoaște mai bine procesul sub aspectul dependențelor dintre variabilele sale cu atît mai simplă și mai economică devine problema conducerii lui, deoarece este

Scanned with OKEN Scanner

suficient să cunoaştem prin măsurare sau determinare un număr mai mic de variabile, la celelalte ajungîndu-se prin deducţie pe baza relaţiilor de dependenţă.

Ex. 10—1. Fie un striper în care un amestec de k hidrocarburi urmează să fie încălzit direct cu abur (prin contact) pentru a îndepărta componenţii uşori în fază vapori, în condiţii de echilibru.

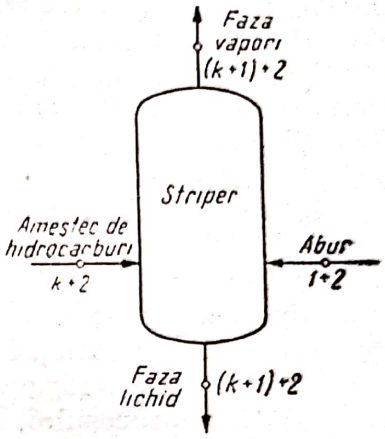


Fig. 10.21.

Fiecare curent de material (hidrocarburi şi abur) care intră sau care iese din aparat este caracterizat (extensiv) prin $k+2$ variabile, unde k este numărul de componenţi ai amestecului ($k-1$ concentraţii, temperatura, presiunea şi debitul).

Urmărind schema din figura 10.21 deducem că numărul total de variabile este:

$$N = 2(k + 3) + k + 2 + 3 = 3k + 11 \qquad (10-2)$$

Numărul de relaţii ce se pot stabili între aceste variabile este $M = 2k + 6$, adică:

$k + 1$ relaţii de conservare pentru cei k componenţi şi abur;

$k + 1$ relaţii de echilibru între componenţii din faza lichid şi componenţii din faza vapori;

1 relaţie de conservare a căldurii;

1 relaţie de conservare a impulsului;

1 relaţie de egalitate a temperaturii celor două faze;

1 relaţie de egalitate a presiunii celor două faze.

$2k + 6$

Rezultă deci că numărul de variabile care trebuie cunoscute pentru a determina starea procesului de stirpare este:

$$F = N - M = 3k + 11 - (2k + 6) = k + 5. \qquad (10-3)$$

Aceste variabile pot fi, de exemplu: $k - 1$ concentraţii, temperatura, presiunea şi debitul amestecului de hidrocarburi şi temperatura, presiunea şi debitul de abur.

Variabila sau funcţia de performanţă. Exprimă eficienţa cantitativă şi/sau calitativă a procesului condus. Se vorbeşte de variabila de performanţă cînd performanţa poate fi exprimată printr-o singură variabilă primară (temperatură, debit, concentraţie ş.a.) sau de funcţia de performanţă cînd aceasta este determinată de două sau mai multe variabile primare (randament, preţ de cost, beneficiu ş.a.), fiind cunoscută şi sub denumirea de funcţie obiectiv, funcţie criterială ş.a.

Variabila sau funcţia de performanţă are o semnificaţie esenţială în cazul optimizării proceselor tehnologice unde urmează să i se asigure o valoare extremală (minimă sau maximă) prin modificarea corespunzătoare a uneia sau a mai multor variabile comandate ale procesului.

Scanned with OKEN Scanner

În legătură cu alegerea şi proprietăţile funcţiei de performanţă merită să subliniem aspectele ce urmează.

Funcţia de performanţă trebuie să exprime de regulă un singur criteriu (indicator) de optimizare şi să nu conţină cerinţe de optimizare contradictorii (un exemplu de cereri contradictorii : calitate maximă cu cheltuieli minime, care sînt incompatibile).

În cazurile în care se impun două sau mai multe cerinţe, numai una urmează a fi exprimată şi extremizată în cadrul funcţiei de performanţă, celelalte urmînd a fi luate în consideraţie sub forma unor restricţii. De pildă, beneficiu maxim şi produse cu calitate cuprinse între anumite limite. De asemenea, în funcţia de performanţă nu se includ termeni constanţi.

Funcţia de performanţă trebuie exprimată prin cel puţin o variabilă comandată a procesului, altfel ea n-ar avea sens şi nici n-ar putea fi extremizată. Dacă nu are un extremum atunci ea trebuie să fie însoţită de cel puţin o restricţie care să confere un sens optimizării — cazul programării liniare [8].

Cele mai uzuale tipuri de funcţii de performanţă sînt:

Funcţia beneficiului, care are expresia:

$$F = \sum_{i=1}^{n} n_i(V_i - C_i), \tag{10—4}$$

unde: V_i reprezintă valoarea specifică a sortimentului i, C_i — cheltuielile specifice de producţie pentru sortimentul i, n_i — numărul unităţilor produse din sortimentul i, n — numărul de sortimente.

Funcţia costului, care are expresia:

$$F = \sum_{i=1}^{n} C_i P_i, \tag{10—5}$$

unde: C_i reprezintă cheltuielile specifice pentru producerea sortimentului i, iar P_i este cantitatea din sortimentul i.

Funcţia calităţii ce poate avea mai multe forme dintre care cea mai uzuală este forma patratică ponderată:

$$F = \sum_{i=1}^{n} p_i(\bar{y}_i - y_i)^2, \tag{10—6}$$

unde: p_i sînt coeficienţi de pondere, $\bar{y}_i$ este valoarea dorită a variabilei optimizate, iar y_i este valoarea efectiv realizată a acesteia.

În cazul programării în regim dinamic, funcţia de performanţă se exprimă de regulă printr-o integrală (sau sumă) a unei funcţii de benficiu, de cost, de calitate sau de timp pe un anumit interval de timp, deci printr-o *funcţională*. Aşa de exemplu, *funcţionala beneficiului*, care urmează a fi maximizată, poate fi pusă sub forma:

$$\max \Phi_b = \max \int_{t_0}^{t} F(y_1, y_2, \ldots, y_m; x_1, x_2, \ldots, x_n; t)\, dt, \tag{10—7}$$

unde: y_1, y_2, ... sînt variabile dependente, iar x_1, x_2, ... — variabile independente.

Scanned with OKEN Scanner

În mod asemănător, *funcționala calității* care urmează a fi minimizată poate avea forma continuă:

$$\min \Phi_c = \min \int_{t_0}^{\infty} \sum_{i=1}^{n} p_i [\bar{y}_i - y_i(t)]^2 \exp [-\alpha(t - t_0)] dt, \quad (10-8)$$

unde α este o constantă pozitivă, sau forma discretă:

$$\min \Phi_c = \min \sum_{k=1}^{\infty} \left\{ \Delta t \sum_{i=1}^{n} p_i [\bar{y}_i - y_i(t_0 + k\Delta t]^2 \right\} \exp [-\alpha k \Delta t,] \quad (10-9)$$

unde $t = k\Delta t$, ambele avînd ca scop asigurarea calității maxime.

Restricțiile proceselor și modelelor. Orice proces tehnologic trebuie astfel condus încît starea lui să se încadreze între anumite limite dinainte stabilite. Exceptînd unele din variabilele de performanță, care în cazul proceselor optimizate urmează a fi menținute la valori extreme, cele mai multe din variabilele de stare trebuie menținute între anumite limite impuse din considerente fizice sau de strategie a conducerii.

Restricțiile de natură fizică sînt impuse în special din considerente de capacitate și securitate. De exemplu, temperatura într-un reactor chimic nu poate depăși o anumită limită care ar favoriza desfășurarea unor reacții nedorite (cocsare, explozii etc.) sau ar pune în pericol instalația și personalul. În general, orice aparat tehnologic sau instalație nu admite debite, presiuni, temperaturi ș.a. mai mari decît cele garantate de constructor.

Restricțiile legate de strategia de conducere sînt determinate atît de calitatea cît și de cantitatea produselor. De pildă, produsele fabricate într-o instalație trebuie să corespundă unor anumiți indici de calitate: puritate, densitate, culoare, perisabilitate etc.

Unele restricții privind cantitatea intervin în cazurile cînd producția trebuie limitată, fie din cauza imposibilităților de desfacere, fie din cauza lipsei mijloacelor de stocare intermediare, fie din cauza supraîncărcărilor capacităților din avalul procesului.

Restricțiile pot fi exprimate sub următoarele forme:

— sub forma „rigidă" a inegalităților, ca de exemplu:

$$\begin{aligned} x_i &\geqslant R_{Ii}, \\ x_i &\leqslant R_{Si}, \end{aligned} \qquad i = 1, 2, \ldots, n \qquad (10-10)$$

unde: R_{Ii} este limita inferioară (poate fi și zero), iar R_{Si} este limita superioară pentru variabila x_i.

Prin introducerea unei variabile auxiliare pozitive

$$z_{Ii} = x_i - R_{Ii} \text{ și respectiv } z_{Si} = R_{Si} - x_i,$$

inegalitățile (10—10) pot fi transformate în egalități, și anume:

$$\begin{aligned} x_i - z_{Ii} &= R_{Ii}, \\ x_i + z_{Si} &= R_{Si}; \end{aligned} \qquad (10-11)$$

Scanned with OKEN Scanner

— sub forma „rigidă" a unor funcţii ale variabilelor de stare şi ale variabilelor de comandă:

$$R_p(y_1, y_2, \ldots, y_m;\ x_1, x_2, \ldots, x_n) \leqslant R_{pS},$$
$$R_p(y_1, y_2, \ldots, y_m;\ x_1, x_2, \ldots, x_n) \geqslant R_{pI}, \qquad (10-12)$$

unde: $p = 1, 2, \ldots, v$, iar R_{pS} şi R_{pI} sînt limita superioară şi respectiv limita inferioară admisibilă.

Prin introducerea unor variabile pozitive auxiliare z, oricare din inegalităţile (10—12) poate fi transformată într-o egalitate de forma:

$$R_p(y_1, y_2, \ldots, y_m;\ x_1, x_2, \ldots, x_n;\ z_p) = 0, \qquad (10-13)$$

iar dacă într-o astfel de relaţie se introduc relaţiile care fac legătura dintre variabilele de stare şi variabilele comandate, adică, modelul matematic al procesului, se obţine o restricţie exprimată numai în funcţie de variabilele de comandă sub forma:

$$R_p(x_1, x_2, \ldots, x_n;\ z_p) = 0; \qquad (10-14)$$

— sub forma nerigidă a unei funcţii de performanţă cu penalizare;

$$F_p = F \pm P, \qquad (10-15)$$

unde penalizarea P urmează să se adauge la valoarea funcţiei de performanţă F dacă aceasta se minimizează sau să se scadă dacă funcţia de performanţă se maximizează.

Un exemplu de funcţie de penalizare este următorul:

$$P = K(X/R)^C, \qquad (10-16)$$

unde K şi C sînt constante pozitive, X este variabila supusă penalizării, iar R este pragul de penalizare.

Putem include în categoria restricţiilor şi condiţiile iniţiale şi/sau finale, adică valorile iniţiale şi/sau finale ale variabilelor de stare $y(t = 0)$, respectiv $y(t = t_f)$, precum şi condiţiile marginale, în cazul modelelor cu parametri distribuiţi.

Tipuri de modele şi premise de elaborare

Diversitatea problemelor de modelat conduce în mod normal şi la o mare varietate de modele. După conţinut, scop urmărit şi după forma de exprimare, modelele pot fi încadrate în următoarele categorii: modele funcţional-procedurale şi modele matematice sau calculatorii.

O deosebită importanţă şi utilitate o au modelele matematice care, la rîndul lor, în funcţie de caracterul procesului modelat pot fi: modele pentru procesele staţionare şi modele pentru procesele dinamice.

De asemenea, în funcţie de premisele elaborării acestor modele, putem vorbi de: *modele deterministe* (elaborate pe baze analitice) şi *modele probabilistice* (elaborate pe baze statistice).

Evident, modelele şi în special modelele matematice pot fi clasificate şi după alte criterii; de exemplu: model matematic precis sau model aproxi-

Scanned with OKEN Scanner

mativ, model matematic complet sau model parţial, model matematic adaptiv sau model invariant ş.a. În cele ce urmează vom prezenta cîteva idei legate de elaborarea şi alegerea celor mai uzuale tipuri de modele.

Modele funcţional-procedurale. Acest gen de modele exprimă ordinea în care urmează să fie executate anumite operaţiuni sau faze ale unui proces, adică întreaga strategie a desfăşurării acestuia pentru diversele situaţii posibile.

Un interes deosebit îl prezintă modelele funcţional-procedurale informaţionale privind colectarea, centralizarea şi prelucrarea informaţiei, cele privind executarea secvenţială a anumitor comenzi, modelele procedurale de verificare şi asigurare a condiţiilor de tehnica securităţii şi protecţia muncii ş.a.

Modelele funcţional-procedurale nu au ca scop principal descrierea unor relaţii cantitative între variabilele procesului modelat, ci descrierea succesiunii operaţiilor de executat în vederea atingerii scopului modelului, inclusiv evaluarea unor relaţii.

Ex. 10—2. Un exemplu de model procedural, care ilustrează succesiunea operaţiilor implicate în pornirea unei pompe centrifugale, este prezentat în figura 10.22.

Se ştie că pentru pornirea unei astfel de pompe trebuie îndeplinite o serie de condiţii „permisive", dintre care menţionăm: *a1* — pompa să fie disponibilă pentru pompajul dorit, *a2* — să dispunem de energie pentru alimentarea motorului, *a3* — să nu fie date interdicţii de pornire, *a4* — să existe încuviinţarea de pornire din partea celui ce conduce instalaţia din care face parte pompa. Toate aceste condiţii trebuie îndeplinite simultan, adică prin conjuncţia: $a1 \cap a2 \cap a3 \cap a4$.

Dacă aceste condiţii sînt îndeplinite se poate trece la pregătirea pompei pentru pornire, în caz contrar se emite un mesaj corespunzător: „Nu sînt îndeplinite condiţiile de pornire".

Pregătirea pompei pentru pornire comportă în principal următoarele măsuri:

— se închide robinetul de pe conducta de refulare;

— se deschid robinetele de pe conductele de aspiraţie şi recirculare;

— se pune în stare de funcţionare manometrul şi debitmetrul sau semnalizorul de debit de pe conducta de refulare şi tahometrul.

Se trece apoi la verificarea acestor condiţii de pornire testînd următoarele stări:

b1 — robinetul de refulare este închis?
b2 — robinetul de aspiraţie este deschis?
b3 — robinetul de recirculare este deschis?
b4 — manometrul de pe refulare funcţionează?
b5 — debitmetrul sau semnalizorul de debit funcţionează?

Dacă toate aceste condiţii sînt îndeplinite ($b1 \cap b2 \cap b3 \cap b4 \cap b5$) se trece la operaţia următoare: conectarea motorului la reţeaua de alimentare; în caz contrar se verifică dacă acest rezultat este primit pentru prima oară. În caz afirmativ se repetă operaţia de pregătire a pompei, iar în caz contrar se emite un mesaj „Pompa nu poate fi pregătită pentru pornire".

Scanned with OKEN Scanner

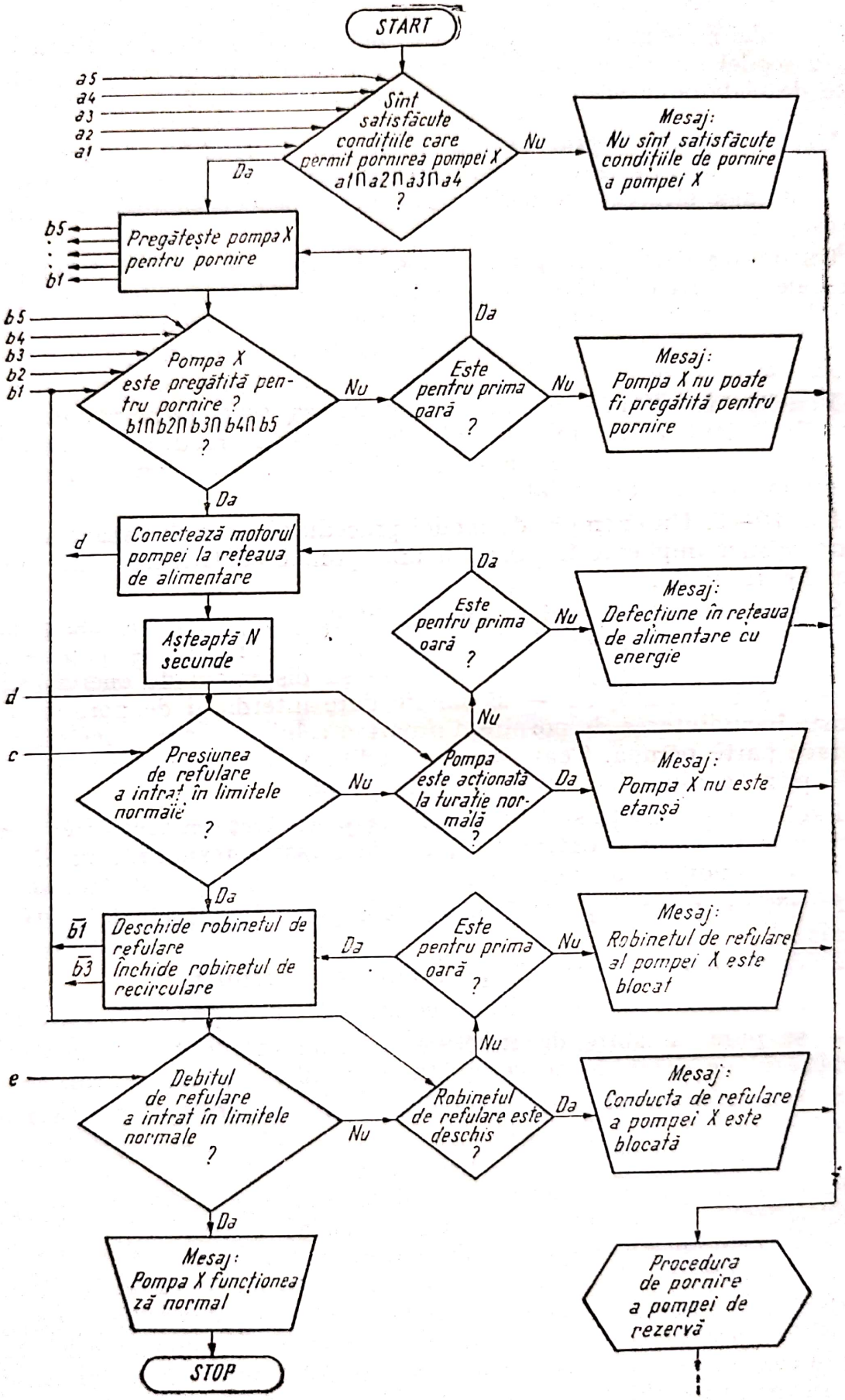


Fig. 10.22.

Scanned with OKEN Scanner

După conectarea motorului la reţea se aşteaptă N secunde pentru demararea acestuia, apoi se verifică dacă presiunea de refulare (variabila *c*) a intrat în limite normale. În caz afirmativ se trece la deschiderea robinetului de refulare, iar în caz contrar se verifică dacă axul pompei se roteşte (variabila *d*). În caz afirmativ se emite un mesaj : „Pompa nu este etanşă", iar în caz contrar se verifică dacă acest rezultat este primit pentru prima oară. În caz afirmativ se repetă operaţia de conectare la reţea, iar în caz contrar se emite mesajul : „Defecţiuni în sistemul de alimentare cu energie".

După deschiderea robinetului de refulare se verifică dacă debitul de refulare (variabila *e*) a intrat în limite normale. În caz afirmativ se emite mesajul final : „Pompa *x* funcţionează normal", iar în caz contrar se testează poziţia robinetului de refulare ; dacă acesta este deschis se emite mesajul „Conducta de refulare a pompei *x* este blocată", în caz contrar verificîndu-se dacă acest rezultat se repetă (în caz afirmativ se emite mesajul „Robinetul de refulare este blocat", iar în caz contrar se repetă operaţia de deschidere a acestui robinet).

Evident că pentru pornirea unei astfel de pompe pot fi concepute şi alte numeroase modele procedurale.

Modele matematice. Modelele matematice sînt constituite dintr-un sistem de relaţii matematice între variabilele procesului modelat cu sau fără restricţii, adică un sistem de ecuaţii şi restricţii care leagă între ele variabilele de stare ale unui proces cu una sau mai multe variabile comandate ale acestuia. Acest tip de modele oferă mari posibilităţi de studiu privind optimizarea procesului modelat, întrucît prin rezolvarea sistemului de ecuaţii în condiţii de restricţii se determină valorile optime ale variabilelor comandate.

Aşa cum am mai precizat, modelele matematice pot fi deductive sau predictive.

Un *model deductiv* este alcătuit dintr-o ecuaţie sau sistem de ecuaţii şi eventuale restricţii care permit deducerea unei variabile nemăsurate pe baza celorlalte variabile măsurate, deci cunoscute. Modelele de acest fel se folosesc atunci cînd în strategia de conducere sau în obiectivele aplicaţiei intervin variabile nemăsurabile ; un exemplu : determinarea coeficientului efectiv de transmitere a căldurii într-un schimbător de căldură, care poate fi dedus cu ajutorul unui model deductiv pe baza debitelor, temperaturilor de intrare şi de ieşire, a căldurii specifice ale celor două medii în interacţiune şi a celorlalţi parametri ai aparatului : suprafaţă de schimb de căldură, pierderile calorice prin izolaţii ş.a.

Un *model predictiv* reprezintă un sistem de relaţii care stabilesc legătura funcţională dintre variabilele dependente ale procesului şi variabilele independente ale acestuia, care permite prevederea răspunsului procesului la modificarea variabilelor de intrare.

Spre deosebire de modelul deductiv care determină legătura dintre variabilele măsurate şi cele nemăsurate, modelul predictiv stabileşte legătura între variabilele dependente şi cele independente ale procesului, indiferent dacă sînt sau nu măsurabile.

Pe baza unui model predictiv s-ar putea deduce, prin simulare, care va fi, de pildă, temperatura (măsurabilă) într-un reactor dacă debitul de reactiv sau compoziţia chimică a acestuia suferă o anumită variaţie cunos-

Scanned with OKEN Scanner

cută. Această predicție este foarte utilă în acțiunea de conducere a procesului, întrucît pe baza ei se pot lua măsuri de corelare a acestei variabile cu alte variabile comandate, astfel încît temperatura să se găsească într-o corelație optimă cu toate variabilele esențiale ale procesului de reacție.

Așa cum am mai arătat, modelele matematice pot fi staționare sau dinamice. *Modelele staționare* descriu relații valabile în regim staționar, pe cînd cele dinamice descriu relații valabile atît în regim staționar cît și în regim tranzitoriu.

Modelul dinamic este necesar în cazurile în care comportarea tranzitorie are o importanță majoră în strategia de conducere, de exemplu în optimizarea dinamică. De altfel, chiar și în cazul conducerii staționare trebuie să se cunoască în suficientă măsură dinamica sistemului pentru a ne asigura că acțiunile de conducere staționară nu contravin unor restricții de ordin dinamic.

Ținînd cont de aceste considerații, urmează să se aleagă acel tip de model care corespunde cel mai bine scopului modelului și resurselor de care se dispune.

După ce tipul de model a fost ales, următorul pas îl constituie elaborarea propriu-zisă a modelului. Un exemplu de formulare a unui model procedural a fost prezentat anterior (v. *Ex. 10—2*). În paragraful următor vom indica modul de elaborare a modelelor matematice deterministe.

Elaborarea modelelor matematice ale proceselor industriale se face de obicei combinînd analiza teoretică a fenomenelor cu analiza și verificarea experimentală. Analiza cu metode teoretice prezintă marele avantaj de a oglindi aspectele și relațiile fundamentale ale procesului urmărit, conferind modelului valabilitate pentru o întreagă clasă de procese similare.

Totuși, nu toate procesele industriale pot fi modelate cu ajutorul metodelor teoretice. Cunoașterea acestor procese este destul de deficitară în unele domenii încît nu este posibilă exprimarea matematică a relațiilor dintre variabilele acestora. Tocmai de aceea modelarea unui proces presupune o amplă activitate de cercetare științifică, teoretică și experimentală.

Premisele modelării deterministe

Punctul de plecare pentru elaborarea modelelor deterministe îl constituie diferitele legi ale științelor naturii: legile conservării masei, energiei și stării de mișcare, legile echilibrului ș.a. De regulă se recurge la un compromis între necesitatea de a dispune de modele riguroase, precise, dar care sînt ample și greu de simulat și nevoia de a dispune de modele cît mai simple și mai ușor de simulat.

Modelele deterministe se reprezintă printr-un sistem de ecuații în care intervin variabile cunoscute și variabile necunoscute. Ca un astfel de sistem cu n variabile necunoscute să fie compatabil este necesar ca acesta să fie alcătuit din n ecuații independente, altfel el este subspecificat sau supraspecificat și deci fără soluție unică.

La stabilirea ecuațiilor trebuie să se acorde atenția cuvenită omogeneității dimensionale a acestora, unitățile de măsură să fie folosite consecvent în cadrul aceluiași sistem de unități, iar acolo unde nevoile impun folosirea

Scanned with OKEN Scanner

unor unităţi de măsură din mai multe sisteme să se introducă factori corecţi de conversie a unităţilor dintr-un sistem în altul.

În cele ce urmează sînt prezentate premisele generale de elaborare a modelelor deterministe.

Ecuaţia de conservare a masei. Principiul conservării masei aplicat unui sistem dinamic deschis conduce la o ecuaţie de bilanţ de forma:

$$\begin{pmatrix}\text{Variaţia în unitatea}\\ \text{de timp a masei sis-}\\ \text{temului}\end{pmatrix} = \begin{pmatrix}\text{Debitul de masă}\\ \text{de intrare în sistem}\end{pmatrix} - \begin{pmatrix}\text{Debitul de masă}\\ \text{de ieşire din sistem}\end{pmatrix} \qquad (10-17)$$

Această ecuaţie arată că variaţia „zestrei" de material (masă) a unui sistem este egală cu diferenţa dintre fluxurile totale de substanţă, de intrare şi de ieşire din acel sistem.

Termenul din stînga egalităţii exprimă „dinamicitatea" sistemului. El este nul pentru regim staţionar şi diferit de zero în regim tranzitoriu. Acest termen exprimă viteza cu care diferenţa dintre debitul masic de intrare şi debitul masic de ieşire conduce la variaţia variabilelor de stare ale sistemului.

Sub aspect matematic, acest termen se exprimă printr-o derivată totală a masei ($\mathrm{d}M/\mathrm{d}t$) dacă sistemul este cu parametri uniform localizaţi, deoarece singura variabilă independentă este timpul şi printr-o derivată parţială dacă sistemul este cu parametri neuniform localizaţi, deoarece în acest caz în afară de variabila timp trebuie să se ia în consideraţie cel puţin o coordonată care să indice valoarea locală a parametrului. În primul caz, ecuaţia de bilanţ se referă la întregul sistem, pe cînd în al doilea caz se referă la un element infinitezimal de volum din sistem.

Ex. 10—3. Fie un recipient de volum constant V în care se află masa M de gaz perfect la presiunea p. În acest recipient intră un amestec gazos ideal cu debitul volumetric D_i şi densitatea ρ_i, iar din acesta amestecul gazos iese cu debitul D şi densitatea ρ. (fig. 10.23). Considerăm că transformările din recipient sînt izoterme.

În aceste condiţii, ecuaţia de bilanţ global capătă forma:

$$\frac{\mathrm{d}M}{\mathrm{d}t} = D_i\rho_i - D\rho. \qquad (10-18)$$

Din ecuaţia de stare a gazelor perfecte ($pV = MRT$) prin derivare se obţine:

$$V\frac{\mathrm{d}p}{\mathrm{d}t} = RT\frac{\mathrm{d}M}{\mathrm{d}t} \qquad (10-19)$$

şi substituind $\mathrm{d}M/\mathrm{d}t$ din (10—19) în (10—18), rezultă:

$$\frac{V}{RT}\frac{\mathrm{d}p}{\mathrm{d}t} = D_i\rho_i - D\rho. \qquad (10-20)$$

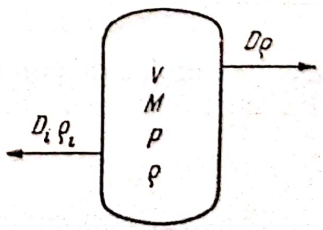


Fig. 10.23.

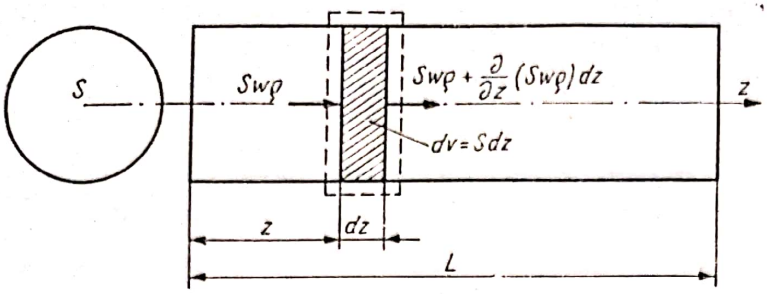


Fig. 10.24.

Ex. 10–4. Fie o conductă de secţiune S prin care curge un fluid în regim izoterm. Viteza liniară de curgere w şi densitatea ρ variază de-a lungul acesteia, adică $w = w(t, z)$ şi $\rho = \rho(t, z)$ aşa cum este arătat în figura 10.24.

Fiind vorba de un sistem cu parametri neuniformi, aplicăm legea conservării asupra unui volum infinitezimal la distanţa z faţă de intrarea în tub. Atunci:

— debitul de masă la intrarea în elementul infinitezimal este $wS\rho$;

— debitul de masă la ieşirea din acest element este $wS\rho + \frac{\partial}{\partial z}(wS\rho)dz$;

— variaţia în unitatea de timp a masei volumului infinitezimal este $\frac{\partial}{\partial z}(ws\rho)dz$;

În acest caz ecuaţia de conservare capătă forma

$$\frac{\partial \rho}{\partial t}(sdz\rho) = ws\rho - \left[ws\rho + \frac{\partial \rho}{\partial z}(ws\rho)dz\right]. \qquad (10-21)$$

Considerînd că secţiunea conductei este constantă şi eliminînd pe dz obţinem:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial z}(\rho w) = 0, \qquad (10-22)$$

adică modelul care corespunde ipotezelor făcute, la care mai trebuie adăugate condiţiile iniţiale $\rho w(t = 0, z) = \rho w$ şi condiţiile la limită (marginale) $\rho w(t, z = 0) = (\rho w)_0$ care permit rezolvarea modelului (10–22).

Ecuaţiile de conservare a masei pe componenţi. În cazul amestecurilor de substanţe supuse unor reacţii chimice sau unor procese de amestecare sau separare se pune problema urmăririi fiecărui component al amestecului în diversele faze ale sistemului, deoarece concentraţiile acestora variază în timp.

Forma generală a ecuaţiei de conservare a componentului i în faza j este:

$$\begin{pmatrix}\text{Variaţia în unitatea de timp a}\\ \text{numărului de moli de component } i\\ \text{din faza } j)\end{pmatrix} = \begin{pmatrix}\text{Debitul molar de component } i\\ \text{din faza } j \text{ ce intră în sistem}\end{pmatrix} -$$

$$\begin{pmatrix}\text{Debitul molar de component } i\\ \text{din faza } j \text{ ce iese din sistem}\end{pmatrix} \mp \begin{pmatrix}\text{Debitul molar de component } i \text{ din}\\ \text{faza } j \text{ transformat şi/sau transfe-}\\ \text{rat în/din alte faze}\end{pmatrix} \qquad (10-23)$$

Fiecărui component al amestecului şi fiecărei faze îi corespunde cîte o astfel de ecuaţie. Totuşi, trebuie avut în vedere că aceste ecuaţii şi ecuaţia

Scanned with OKEN Scanner

de conservare globală pentru o fază nu sînt independente, deoarece bilanţul global rezultă din suma bilanţurilor pe componenţi ai fazei respective. De aceea, de regulă, se folosesc $n - 1$ ecuaţii de conservare pe component şi o ecuaţie de conservare globală pentru fiecare fază.

Ex. 10—5. Fie un reactor cu amestecare perfectă în care are loc o relaţie chimică izotermă omogenă de ordinul I de forma:

$$A \xrightarrow{k} B.$$

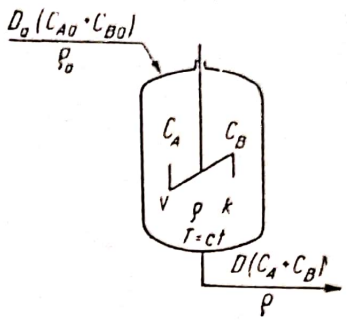


Fig. 10.25.

Debitul în volume al amestecului de intrare este D_0, iar debitul amestecului rezultat la ieşire este D. Concentraţiile celor doi componenţi A şi B la intrarea în reactor sînt C_{A_0} şi respectiv C_{B_0}, iar concentraţiile acestor componenţi în amestecul de ieşire sînt C_A, respectiv C_B. Constanta vitezei de reacţie este k, iar volumul de reacţie este V. (fig. 10.25).

Fiind vorba de un sistem cu parametri uniformi vom face bilanţul unuia din componenţi, de pildă componentul A pe întregul sistem:

— debitul de intrare în moli este: D_0C_{A0};

— debitul de ieşire în moli este: DC_A;

— viteza de transformare a componentului A în component B este: $-VkC_A = VkC_B$;

— variaţia în unitatea de timp a numărului de moli de component A din sistem este $\frac{d}{dt}(VC_A)$.

În aceste condiţii, cele două ecuaţii de conservare ale componenţilor sînt:

$$\begin{aligned} \frac{d}{dt}(VC_A) &= D_0C_{A_0} - DC_A - VkC_A, \\ \frac{d}{dt}(VC_B) &= D_0C_{B_0} - DC_B + VkC_A. \end{aligned} \qquad (10—24)$$

Concentraţiile celor doi componenţi se referă la densitatea ρ care se calculează din relaţia:

$$\rho = M_AC_A + M_BC_B, \qquad (10—25)$$

unde M_A şi M_B sînt masele moleculare ale celor doi componenţi.

Ca ecuaţii de conservare se pot utiliza cele două ecuaţii de conservare ale componenţilor sau una din acestea şi ecuaţia de conservare globală de forma (10—17).

Ex. 10—6. Fie un reactor tubular în care are loc o reacţie izotermă omogenă de tipul $A \xrightarrow{k} B$ de ordinul I, la care gradienţii radiali pot fi neglijaţi, existînd numai gradient axial de viteză, concentraţie şi densitate, adică gradient provocat de curgere (convectiv) şi gradient provocat de diferenţe de concentraţie (difuzional).

Fluxul de difuzie a componentului A în direcţia axială poate fi stabilit de relaţia lui Fick sub forma:

$$-\mathfrak{D}_A \frac{\partial C_A}{\partial z}, \left[\frac{\text{moli de } A}{\text{timp, unitate de supraf.}}\right]$$

unde $\mathfrak{D}_A$ este coeficientul de difuzie [suprafaţa/timp].

Scanned with OKEN Scanner

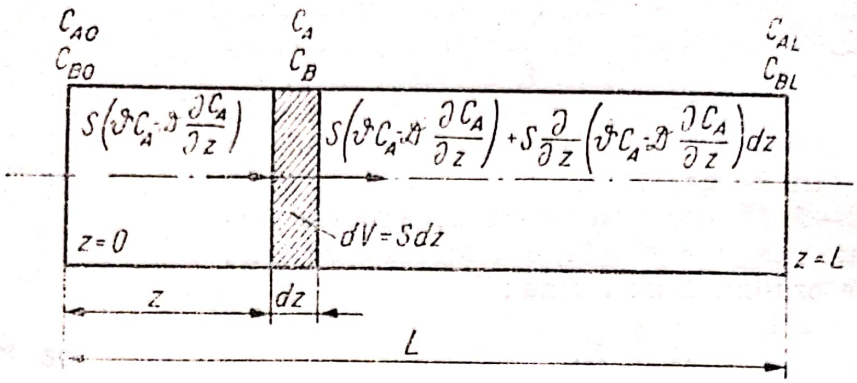


Fig. 10.26.

În vederea stabilirii ecuației de conservare a componentului A asociată unui volum infinitezimal la distanța z față de intrarea în recator (fig. 10.26) determinăm:

— debitul molar de component A (convectiv și difuzional) la intrarea în volumul infinitezimal, care este:

$$wSC_A - \mathcal{D}_A S\frac{\partial C_A}{\partial z},$$

unde cu S s-a notat secțiunea transversală a tubului;

— debitul în moli de component A la ieșirea din volumul infinitezimal:

$$wSC_A - \mathcal{D}_A S\frac{\partial C_A}{\partial z} + \frac{\partial}{\partial z}\left(wSC_A - \mathcal{D}_A S\frac{\partial C_A}{\partial z}\right)dz;$$

— viteza de transformare a componentului A în interiorul volumului infinitezimal: $-kC_A S\,dz$;

— variația în unitatea de timp a numărului de moli de component A în volumul infinitezimal:

$$\frac{\partial}{\partial t}(SC_A\,dz).$$

Grupînd termenii conform relației (10—23) și simplificînd în mod corespunzător obținem:

$$\frac{\partial C_A}{\partial t} = -\frac{\partial}{\partial z}wC_A - kC_A + \frac{\partial}{\partial z}\left(\mathcal{D}_A\frac{\partial C_A}{\partial z}\right). \qquad (10—26)$$

Termenul din stînga egalității exprimă efectul global al variației în timp a concentrației, iar termenii din dreapta: efectul convecției, efectul reacției și respectiv efectul difuziei axiale.

Ecuația de conservare a energiei. Aplicat unui sistem termodinamic deschis, principiul conservării energiei (prima lege a termodinamicii) con-

Scanned with OKEN Scanner

duce la o ecuaţie de forma:

$$\begin{pmatrix}\text{Variaţia în unitatea de timp a}\\ \text{energiei interne, cinetică şi po-}\\ \text{tenţială ale sistemului}\end{pmatrix} = \begin{pmatrix}\text{Fluxul total de energie}\\ \text{internă, cinetică şi potenţială}\\ \text{la intrarea în sistem}\end{pmatrix} -$$

$$- \begin{pmatrix}\text{Fluxul total de energie internă,}\\ \text{cinetică şi potenţială la ieşirea}\\ \text{din sistem}\end{pmatrix} \pm \begin{pmatrix}\text{Fluxul total de energie}\\ \text{dezvoltată sau consumată în}\\ \text{sistem prin reacţie}\end{pmatrix} -$$

$$- \begin{pmatrix}\text{Fluxul total de energie consuma-}\\ \text{tă în sistem pentru deplasare}\\ \text{şi variaţie de presiune şi volum}\end{pmatrix} \qquad (10-27)$$

Această formă generală capătă diverse forme particulare care în cele mai multe cazuri se reduc la relaţii de bilanţ caloric, fie pe întregul sistem dacă acesta este omogen, fie pe diversele faze ale acestuia dacă sistemul este heterogen.

Ex. 10—7. Fie un reactor chimic în care are loc o reacţie chimică exotermă de tipul $A \xrightarrow{k} B$, înzestrat cu o serpetină de răcire prin care se poate asigura transferul fluxului de căldură Q.

Produsul reactant intră în reactor cu debitul D_0, presiunea p_0, densitatea ρ_0 şi temperatura T_0. Reacţia are loc la temperatura T şi presiunea p. Produsul rezultat are densitatea ρ şi este evacuat cu debitul D.

Fluxul de căldură degajată de reacţie este:

$$Q_R = -q_R V C_A k,$$

unde: q_R este căldura de reacţie, V — volumul de reacţie şi k — constanta vitezei de reacţie.

Particularizînd ecuaţia (10—27) la cazul de faţă obţinem:

$$\frac{d}{dt}[(U + C + P)V\rho] = D_0\rho_0(U_0 + C_0 + P_0) - D\rho(U + C + P) + (Q_R - Q) - (L + Dp - D_0p_0)E, \qquad (10-28)$$

unde U este energia internă specifică (raportată la unitatea de masă), kcal/kg, C — energia cinetică specifică, P — energia potenţială specifică, L — lucrul util (de deplasare), iar E — echivalentul caloric al lucrului mecanic (în sistemul MKFS $E = 2{,}343 \times 10^{-3}$ kcal/kgf·m).

În cazul de faţă, lucrul mecanic de deplasare este nul. Dacă vitezele de intrare, respectiv de ieşire din reactor, şi diferenţa de nivel a conductelor de intrare şi de ieşire sînt neînsemnate, putem neglija termenii care ţin cont de energia cinetică şi potenţială, astfel că relaţia (10—28) devine:

$$\frac{d}{dt}(UV\rho) = D_0\rho_0U_0 - D\rho U + Q_R - Q - D\rho\left|\frac{p}{\rho}\right|E + D_0\rho_0\left|\frac{p_0}{\rho_0}\right|E = D_0\rho_0(U_0 + p_0v_0E) - D\rho(U + pvE) + Q_R - Q, \qquad (10-29)$$

unde v este volumul specific ($v = 1/\rho$).

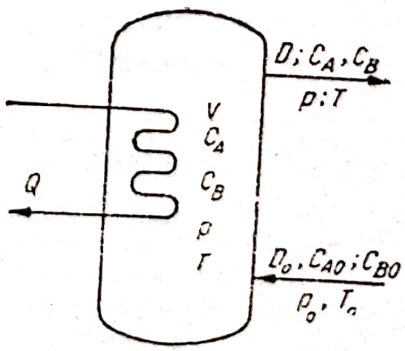


Fig. 10.27.

În locul termenului U (energie internă) putem utiliza termenul H (entalpie) cunoscînd că:

$$H = U + pvE. \tag{10—30}$$

Cu această notație relația (10—29) devine:

$$\frac{d}{dt}(\rho VU) = D_0\rho_0 H_0 - D\rho H - Q + q_R VkC_A. \tag{10—31}$$

În cazurile în care $pV \ll U$ relația (10—31) poate fi înlocuită prin relația:

$$\frac{d}{dt}(\rho VH) = D_0\rho_0 H_0 - D\rho H - Q + \rho_R VkC_A, \tag{10—32}$$

care la rîndul ei poate fi înlocuită cu alte relații pentru diverse cazuri particulare.

Ecuațiile de conservare a stării de mișcare. Principiul conservării mișcării, exprimat prin legea a II-a a lui Newton, aplicată la un sistem cu masă variabilă conduce la o ecuație de forma:

$$\frac{d}{dt}(Mw_i) = \sum_{j=1}^{n} F_{ij}, \tag{10—33}$$

unde M este masa sistemului, w_i — viteza* în direcția i, iar F_{ij} — componenta forței j ce acționează în direcția i.

Această relație arată că variația impulsului (Mw) în direcția i este egală cu suma algebrică a forțelor ce acționează în acea direcție și exprimă un echilibru al forțelor. Ea constituie baza de plecare pentru stabilirea tuturor ecuațiilor de mișcare mecanică (Lagrange, Hamilton ș.a.).

Dacă masa M este constantă, relația (10—33) capătă binecunoscuta formă:

$$Ma_i = \sum_{i=1}^{n} F_{ij}, \tag{10—34}$$

unde a_i este accelerația în direcția i.

Mișcarea mecanică poate fi descompusă pe trei direcții, prin urmare se pot stabili trei ecuații de echilibru al forțelor, corespunzător celor trei direcții.

Ex. 10—8. Fie o conductă pentru transportul succesiv a două produse diferite cu densitățile ρ_1 și ρ_2.

Pentru ca produsele respective să nu se „contamineze" reciproc, ele sînt separate cu un piston mobil dintr-un material elastic ce se deplasează prin conductă odată cu interfața dintre cele două produse. Masa acestui piston este neglijabilă în raport cu masa produselor din conductă.

---

* Se au în vedere viteze mici în raport cu viteza luminii.

Scanned with OKEN Scanner

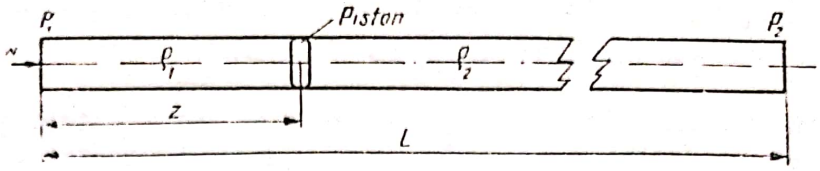


Fig. 10.28.

Pentru simplificare, considerăm că densităţile celor două fluide sînt constante şi că forţa de frecare a fluidelor cu conducta este proporţională cu pătratul vitezei de curgere (fig. 10.28).

Să considerăm că pistonul se află la distanţa z faţă de intrarea în conductă (fig. 10—28). În acest caz avem:

— forţa activă de presiune $F_a = S(p_1 - p_2)$;

— forţa de frecare $F_f = F_{f1} + F_{f2} = k_{f1} z w^2 + k_{f2}(L - z)w^2$, unde k_{f1} şi k_{f2} sînt coeficienţi de frecare;

— masa fluidelor conţinute de conductă $M_z = Sz\rho_1 + S(L - z)\rho_2$.

Grupînd termenii conform relaţiei (10—33) obţinem în final:

$$S \frac{\mathrm{d}}{\mathrm{d}t} (z\rho_1 + (L - z)\rho_2)w = S(p_1 - p_2) - [k_{f1} z + k_{f2}(L - z)]\, w^2. \quad (10—35)$$

Ecuaţiile de transport. Legile conservării masei, energiei şi mişcării prezentate mai înainte sînt în acelaşi timp legi care descriu transportul de masă, energie şi mişcare. În relaţiile stabilite transferul este evidenţiat sub forma unui flux (viteză de transfer raportată la unitatea de suprafaţă) proporţional cu o forţă motoare (un gradient de concentraţie, temperatură sau viteză) care reflectă o proprietate fizică a sistemului.

În baza acestei observaţii se poate stabili o ecuaţie generală de transport care să capete diverse forme particulare pentru a pune în evidenţă legile de conservare expuse mai înainte. În acest scop să luăm în consideraţie un sistem S delimitat de o hipersuprafaţa A la care participă φ_i faze şi din care rezultă φ_e faze (fig. 10.29).

Să presupunem că în cadrul acestui sistem au loc următoarele tipuri de procese: convecţie, difuzie, transfer între faze, precum şi procese de generare sau consum (de exemplu: reacţii chimice cu degajare sau consum de energie). Datorită acestor procese numărul şi concentraţia componenţilor celor φ_e faze ce ies din sistem sînt diferite de numărul şi concentraţia componenţilor celor φ_i faze ce intră în sistem.

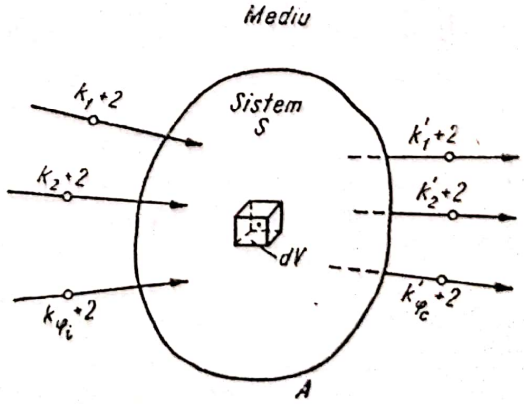


Fig. 10.29.

Fie z o mărime care poate să exprime o cantitate de masă, de energie sau de impuls raportată la volumul întregului sistem sau la un volum infini-

tezimal al sistemului pe care o vom denumi densitate generalizată, Γ. Folosind această variabilă generalizată putem scrie:

$$\begin{pmatrix}\text{Variaţia în unitatea de timp a}\\ \text{densităţii generalizate}\end{pmatrix} = \begin{pmatrix}\text{Schimbul între sistem şi mediu}\\ \text{prin difuzie în unitatea de timp}\end{pmatrix} +$$

$$+ \begin{pmatrix}\text{Schimbul între sistem şi mediu}\\ \text{prin convecţie în unitatea de timp}\end{pmatrix} + \begin{pmatrix}\text{Schimb interfazic în unitatea}\\ \text{de timp}\end{pmatrix} \pm$$

$$\pm \begin{pmatrix}\text{cantitate generată sau consuma-}\\ \text{tă în sistem în unitatea de timp}\end{pmatrix}. \qquad (10-36)$$

În funcţie de scopul urmărit ecuaţia (10—36) poate fi particularizată pentru schimbul de masă, pentru schimbul de energie sau pentru schimbul de impuls, pentru întregul sistem, pentru o anumită fază a sistemului sau chiar pentru un anumit component dintr-o anumită fază.

Înlocuind termenii din ecuaţia (10—36) cu expresiile lor matematice obţinem o ecuaţie generalizată de forma:

$$\frac{\partial \Gamma}{\partial t} = \operatorname{div}\,(k \operatorname{grad} \Gamma) - \operatorname{div}\,\Gamma\bar{w} - a\alpha\Delta\Gamma - G. \qquad (10-37)$$

unde k este un coeficient generalizat de difuzie [m²/s], $\bar{w}$ — vectorul vitezei lineare, a — suprafaţa specifică de transfer [m²/m³], $\Delta\Gamma$ — diferenţa densităţii generalizate Γ, iar G — debitul sursei sau consumatorului din interiorul sistemului.

Ex. 10—9. Fie un sistem alcătuit dintr-un vas cu lichid de volum V în care se află scufundat un termometru de volum V_T şi masa M_T foarte mică în comparaţie cu masa M a lichidului (fig. 10.30).

Se cere modelul dinamic care să exprime dependenţa dintre temperatura θ_T a termometrului şi temperatura θ a lichidului considerînd că întrucît $M_T \ll M$, $\theta = ct$.

În acest caz avem un sistem cu parametri uniformi. Ecuaţia (10—37) urmează a fi folosită cu ecuaţie de bilanţ caloric pentru faza solid (termometru).

Densitatea generalizată este conţinutul caloric specific al termometrului (entalpia unităţii de volum), adică:

$$\Gamma = \frac{M_T c_{pT} \theta_T}{V_T} = \rho_T c_{pT} \theta_T.$$

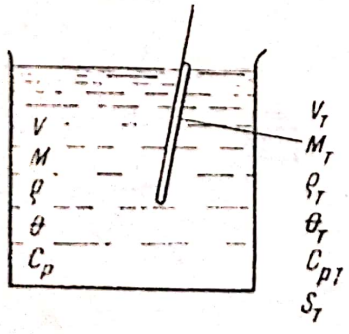


Fig. 10.30.

Schimbul de căldură prin convecţie şi conductibilitate cu mediul exterior este nul, iar corpul termometrului are aceeaşi temperatură în toate punctele sale; deci grad $\Gamma = 0$ şi div $\Gamma\bar{w} = 0$. De asemenea, în sistem nu sînt nici surse nici consumatori de căldură, deci $G = 0$. Singurul transfer de căldură are loc între lichid şi termometru, forţa motrice fiind diferenţa de temperatură $\theta_T - \theta$.

Scanned with OKEN Scanner

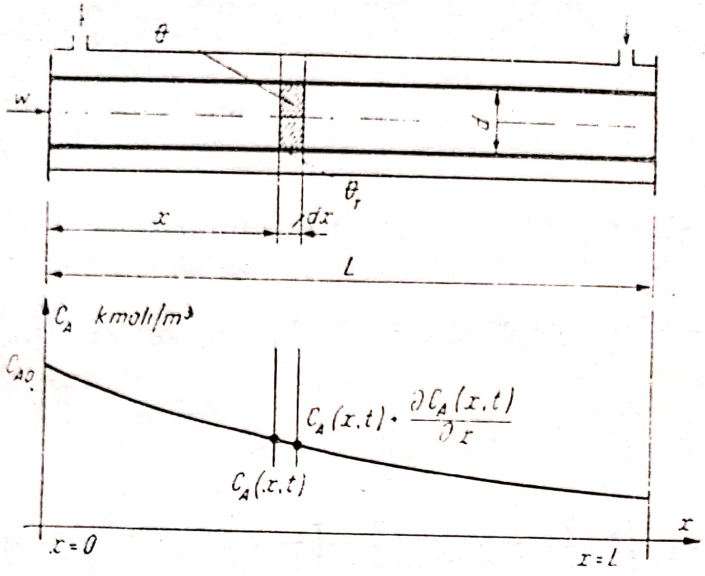


Fig. 10.31.

În aceste condiţii ecuaţia (10—37) capătă forma:

$$\frac{\partial}{\partial t}(\rho_T C_{pT}\theta_T) = -\frac{S_T}{V_T}K(\theta_T - \theta),$$

unde S_T este suprafaţa exterioară a termometrului, iar K coeficientul de transfer de căldură.

Regrupînd termenii constanţi se obţine:

$$\frac{V_T\rho_T C_{pT}}{S_T K}\frac{d\theta_T}{dt} + \theta_T = \theta. \qquad (10—38)$$

Ex. 10—10. Se cere modelul dinamic al unui reactor chimic de tip tubular (fig. 10—31) cu manta pentru răcire, în care are loc reacţia exotermă $A \xrightarrow{k} B$ ce se desfăşoară cu viteza:

$$r = kC_A e^{-\frac{E}{R\theta}}.$$

Pentru simplificare se presupune că viteza, concentraţiile şi temperatura reactanţilor au numai gradienţi axiali (cei radiali fiind neglijabili) şi că pierderile de presiune ale celor două medii (reactant şi mediu de răcire) sînt de asemenea, neglijabile. În fine, se neglijează şi capacitatea calorică a peretelui despărţitor al celor două medii precum şi variaţia cu temperatura a căldurii specifice C_p, a densităţii ρ şi a căldurii de reacţie.

În acest caz ecuaţia (10—37) urmează să fie folosită ca ecuaţie de conservare a masei unuia din componenţi, de exemplu pentru componentul A, şi ca ecuaţie de conservare a căldurii mediului reactant, ambele raportîndu-se la un volum infinitezimal ΔV.

Pentru prima ecuaţie variabila Γ exprimă concentraţia componentului A, adică:

$$\Gamma = \frac{N_A}{V} = C_A,$$

Scanned with OKEN Scanner

unde N_A este numărul de moli din volumul infinitezimal ΔV, iar pentru cea de a doua ecuaţie exprimă entalpia raportată la unitatea de volum, adică :

$$\Gamma = \frac{MC_p\theta}{\Delta V} = \rho C_p \theta,$$

unde M, C_p, θ şi ρ reprezintă masa medie a amestecului reactant din volumul ΔV, căldura specifică, temperatura şi respectiv densitatea medie a amestecului reactant.

Cu aceste notaţii cele două ecuaţii obţinute din ecuaţia (10—37) devin :

$$\frac{\partial C_A}{\partial t} = \frac{\partial}{\partial x}\left(\mathfrak{D}\frac{\partial C_A}{\partial x}\right) - w\frac{\partial C_A}{\partial x} - kC_A e^{-\frac{E}{R\theta}}, \qquad (10—39)$$

$$\rho C_p \frac{\partial \theta}{\partial t} = \frac{\partial}{\partial x}\left(\lambda \frac{\partial \theta}{\partial x}\right) - \rho C_p w \frac{\partial \theta}{\partial x} + \left(-q_R k C_A e^{-\frac{E}{R\theta}} + \frac{1}{d}\alpha(\theta - \theta_T)\right),$$

unde : $\mathfrak{D}$ este coeficientul de difuzie, w — viteza liniară a amestecului reactant, λ — coeficientul de conductibilitate termică, α — coeficientul global de transmitere a căldurii, iar q_R — căldura de reacţie.

La aceste ecuaţii trebuie adăugate condiţiile iniţiale :

$$C_A(x, 0) = C_A(x); \quad \theta(x, 0) = \theta(x), \qquad (10—40i)$$

precum şi condiţiile marginale :

$$C_A(0, t) = C_A(t); \quad \theta(0, t) = \theta(t). \qquad (10—40m)$$

Dacă nu se impun restricţii cu privire la limitele de variaţie ale variabilelor menţionate, modelul matematic este alcătuit din ecuaţiile (10—39), condiţiile iniţiale (10—40*i*) şi condiţiile marginale (10—40*m*).

Prelucrarea şi perfecţionarea modelelor

Ca să poată fi simulat pe un calculator numeric modelul matematic trebuie adus la o formă adecvată calculatorului. De exemplu, ecuaţiile diferenţiale urmează să fie transformate în ecuaţii cu diferenţe care se pretează rezolvării numerice şi să se aleagă acele metode numerice de rezolvare care asigură o cît mai bună convergenţă a soluţiilor.

La elaborarea modelelor pentru optimizarea proceselor trebuie să se ţină seama de metoda de optimizare care va fi adoptată, întrucît aceasta condiţionează şi forma modelului. De exemplu, una va fi forma modelului dacă se adoptă o metodă de optimizare staţionară şi alta dacă se adoptă o metodă de optimizare dinamică.

Oricît de bun ar fi la un moment dat un model, trebuie să existe o preocupare permanentă pentru îmbunătăţirea acestuia. Actualizarea modelului şi diminuarea neconcordanţei dintre acesta şi procesul modelat trebuie să facă parte din tactica şi strategia de utilizare a modelului, mai ales cînd acesta este folosit pentru optimizare.

Modelul poate fi perfecţionat pe baza datelor acumulate în timp şi prelucrate de un calculator pe baza unui program adecvat. Astfel calcula-

Scanned with OKEN Scanner

torul poate să consemneze în mod sistematic discrepanţele dintre răspunsul pronosticat de model şi cel real dat de proces ca pe această bază să perfecţioneze (adapteze) modelul.

În fine există posibilitatea ca pe baza metodelor avansate de autoinstruire, calculatorul să fie în măsură să conducă procese pe baza unui model optim elaborat din experienţa proprie.

Noţiuni referitoare la identificarea proceselor

Identificarea este acea componentă a activităţii de analiză a sistemelor prin care se determină parametrii reprezentativi ai unui proces sau instalaţii, în funcţie de valorile variabilelor de intrare şi ieşire ale sistemului, în vederea reprezentării lui printr-un model *echivalent* din punctul de vedere al izomorfismului. În acest scop este nevoie să se stabilească de la început clasa de modele luată în consideraţie, clasa de semnale de intrare considerate şi noţiunea de echivalenţă.

În cele mai multe cazuri, echivalenţa se exprimă printr-un criteriu de eroare (abatere) sau penalizare care sînt funcţii sau funcţionale ale ieşirilor procesului şi modelului, adică:

$$E = E(x, y, y_M), \qquad (10-41)$$

unde x este variabila de intrare, y — variabila de ieşire a procesului modelat, iar y_M — variabila de ieşire a modelului.

Problema identificării poate fi pusă şi rezolvată în foarte multe feluri, în funcţie de informaţia apriorică de care dispunem şi de scopul şi exigenţa identificării.

De obicei, identificarea presupune cunoaşterea unui minimum de informaţie apriorică asupra procesului modelat, îndeosebi informaţia referitoare la forma şi structura modelului.

În figura 10—32 este sugerată legătura dintre informaţia apriorică şi informaţia posteriorică în procesul de identificare. Ramura superioară ilustrează procesul de elaborare a formei şi structurii modelului pe baza

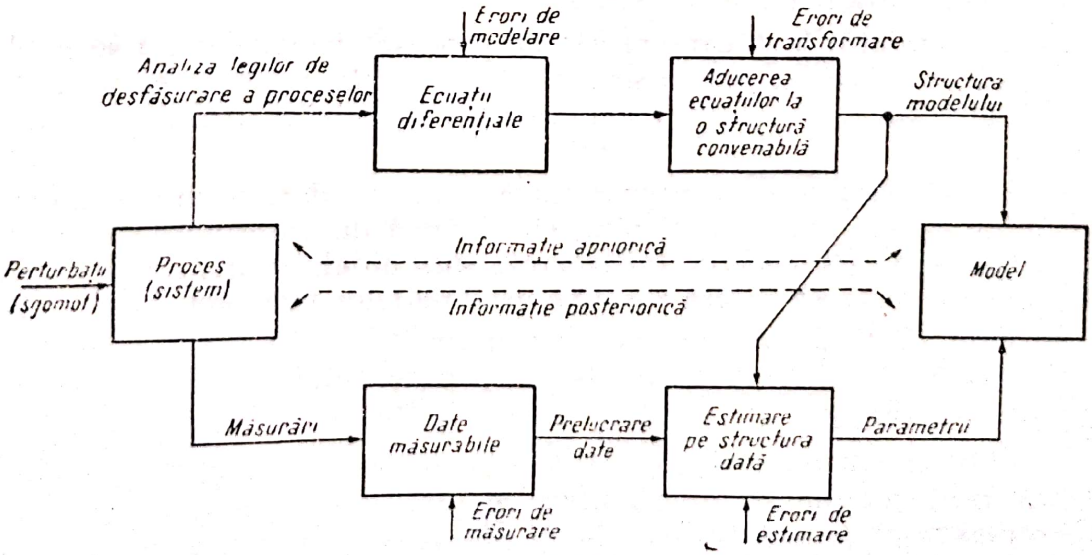


Fig. 10.32.

Scanned with OKEN Scanner

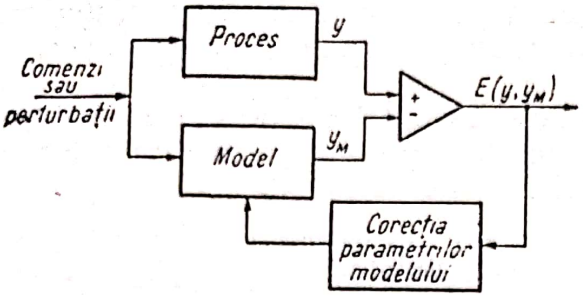


Fig. 10.33.

analizei legilor după care se desfășoară procesul identificat, iar ramura inferioară ilustrează procedura de estimare a parametrilor modelului bazată pe măsurarea variabilelor de intrare și ieșire și pe prelucrarea datelor astfel obținute. Atît pe o ramură cît și pe cealaltă erorile sînt inerente.

În cazul proceselor cu mai multe variabile, identificarea este dificilă deoarece de regulă fiecare variabilă de ieșire este determinată de mai multe variabile de intrare. În acest caz pentru exprimarea variabilelor de intrare și de ieșire se folosesc vectori ; $\mathbf{x} = \mathbf{x}(x_1, x_2, \ldots, x_m, t)$, respectiv $\mathbf{y} = \mathbf{y}(y_1, y_2, \ldots, y_n, t)$.

Problema centrală a identificării proceselor constă în estimarea parametrilor modelului, adică în determinarea experimentală a valorilor parametrilor ce caracterizează comportarea procesului modelat în ipoteza că structura modelului este dinainte stabilită. Aceasta se poate face în felul următor : asupra procesului și modelului se aplică același semnal de intrare x ; se compară semnalul de ieșire al procesului, y, afectat de zgomot cu semnalul de ieșire al modelului, y_m, și se corectează de așa manieră parametrii modelului încît diferența dintre cele două semnale de ieșire să fie minimă (fig. 10—33).

Drept criteriu pentru îmbunătățirea parametrilor modelului servește funcționala abaterii care poate fi exprimată sub forma :

$$min\ \mathbf{E}(\mathbf{y}, \mathbf{y}_M) = min \int_0^T (\mathbf{y} - \mathbf{y}_M)^2 dt. \qquad (10—42)$$

Prin urmare identificarea poate fi tratată ca o problemă de optimizare, atît în ceea ce privește structura modelului cît, mai ales, în ce privește determinarea parametrilor acestuia.

Dacă proprietățile și caracteristicile procesului modelat nu se modifică în timp, identificarea se face o singură dată. Dacă însă acestea se modifică în timp, identificarea pentru reactualizarea modelului urmează să se facă periodic.

Estimarea parametrilor se poate face atît după date obținute în condiții normale de exploatare a procesului, cît și după date obținute ca urmare a unor perturbații artificiale de tip treaptă, sinusoidal, rampă ș.a. În legătură cu aceasta urmează să se dea răspuns următoarelor întrebări de procedură :

— cum să se planifice experimentul ?
— cum să se analizeze rezultatele experimentului ?
— ce perturbații să se aplice pentru a obține efectul optim ?
— care să fie perioada de testare și durata experimentului ? ș.a.

Diversele posibilități de abordare și rezolvare a acestor probleme se diferențiază după *procedura de realizare*. Totuși în principiu aceste proceduri pot fi încadrate în două categorii :

Scanned with OKEN Scanner

— *proceduri de determinare directă* a parametrilor modelului pe baza unor relaţii prestabilite şi a unui set de date experimentale, prelucrate analitic într-o singură etapă, adică proceduri directe sau explicite;

— *proceduri de determinare (optimizare) iterativă*, analitică sau experimentală a parametrilor modelului, astfel încît prin corecţii succesive caracteristicile modelului să devină din ce în ce mai apropiate (la limită identice) de caracteristicile procesului identificat.

Deosebirea dintre cele două categorii de proceduri constă deci în modul şi în posibilităţile de a determina parametrii modelului şi poate fi înţeleasă mai uşor din explicaţiile de mai jos.

Să presupunem că modelul are aceeaşi structură ca şi procesul modelat, şi că la ieşirea procesului se obţine semnalul

$$\mathbf{y}(t) = f(\mathbf{x},\ \mathbf{a},\ \mathbf{v}), \tag{10—43}$$

iar la ieşirea modelului, semnalul:

$$\mathbf{y}_M(t) = f(\mathbf{x},\ \alpha,\ \mathbf{v}), \tag{10—44}$$

unde: $\mathbf{a}$ este vectorul m dimensional al parametrilor procesului; α — vectorul m dimensional al parametrilor modelului; $\mathbf{x} = \mathbf{x}(t)$ — semnalul de intrare al procesului şi modelului şi $\mathbf{v} = \mathbf{v}(t)$ — perturbaţii.

Să admitem că măsura „echivalenţei" procesului şi modelului este funcţionala abaterii (erorii):

$$\mathbf{E}[\mathbf{y},\ \mathbf{y}_M,\ \alpha] = \int_{t-T}^{t} [\mathbf{y}(t) - \mathbf{y}_M(t),\ \alpha]\mathrm{d}t. \tag{10—45}$$

Pentru determinarea parametrilor α care minimizează eroarea se poate proceda în două moduri:

1° Calculînd derivatele erorii în raport cu parametrii α şi anulînd derivatele, adică:

$$\frac{\partial \mathbf{E}}{\partial \alpha_i} = 0;\ i = 1,\ 2,\ \ldots\ m, \tag{10—46}$$

de unde se obţine un sistem de m ecuaţii cu m parametri necunoscuţi $(\alpha_1,\ \alpha_2,\ \ldots\ \alpha_m)$, care poate fi rezolvat în raport cu α, adică o procedură din categoria a întîia, care conduce la un rezultat după un număr finit de operaţii matematice executate într-o singură etapă.

Această procedură este realizabilă numai dacă modelul este exprimat sub forma unor expresii matematice determinate.

2° Urmărind evoluţia derivatelor $\partial \mathbf{E}/\partial \alpha_i$ şi modificînd parametrii α astfel încît să obţinem o valoare din ce în ce mai mică a derivatelor, adică urmărind tendinţa de diminuare a erorii în raport cu parametrul α.

O astfel de procedură este posibilă numai dacă există mijloace tehnice de modificare a modelului şi de măsurare a efectului şi dacă analiza acestor derivate este folosită cu scopul acordării modelului (fizic sau teoretic), adică o abordare cu o procedură din categoria a doua. În acest caz, teoretic, soluţionarea se obţine după un număr infinit de operaţii elementare, alternînd optimizarea parametrilor modelului cu analiza efectului obţinut.

Scanned with OKEN Scanner

Precizia identificării cu procedurile din prima categorie depinde sensibil de calitatea structurii modelului. Precizia identificării prin procedurile din categoria a doua este mult mai bună, deoarece aceste proceduri dau posibilitatea îmbunătăţirii parametrilor şi chiar a structurii modelului prin legătura inversă ce pune în valoare o însemnată cantitate de informaţie obţinută pe această cale. Mai mult decît atît, în acest caz pot fi folosite la scară reală de timp chiar şi rezultatele intermediare, ceea ce înseamnă mult, mai ales dacă comportarea procesului identificat se modifică în timp. De aceea se şi recomandă ca această procedură să fie adoptată pentru identificarea proceselor cu comportare variabilă.

În ultimă instanţă problema identificării se formulează ca o problemă de minimizare şi anume, minimizarea funcţionalei:

$$\mathbf{E}[\mathbf{y}(t, \mathbf{a}) - \mathbf{y}_M(t, \alpha)] \qquad (10-47)$$

sau a speranţei matematice a funcţionalei:

$$\mathcal{E}[\mathbf{E}(\mathbf{a} - \alpha)], \qquad (10-48)$$

unde $\mathcal{E}$ este speranţa matematică a funcţiei $\mathbf{E}(\mathbf{a} - \alpha)$.

În literatura de specialitate sînt descrise diverse proceduri de identificare care se diferenţiază după criteriul de optimalitate ales şi după informaţia apriorică pe care se bazează.

Metode de estimare a modelelor

Metoda regresiei. Metoda estimării prin regresie se foloseşte pentru determinarea coeficienţilor modelelor proceselor staţionare după un set de date ridicate experimental.

Fie un proces liniar staţionar cu o singură variabilă de ieşire y şi cu m variabile de intrare x_j, $j = 1, 2, \ldots, m$ de la care s-a obţinut setul de n date $(y_1, x_{11}, x_{12}, \ldots, x_{1m})$, $(y_2, y_{21}, x_{22}, \ldots, x_{2m})$, $\ldots$, $(y_n, x_{1n}, x_{2n}, \ldots, x_{mn})$.

Presupunem că se cere ca procesul să fie aproximat de către un model liniar de forma:

$$y_M = \alpha_0 + \sum_{j=1}^{m} \alpha_j x_j, \qquad (10-49)$$

unde y_M este variabila de ieşire a modelului, iar x_j — variabilele de intrare.

Pentru ca modelul să descrie corect procesul este necesar să se determine valorile coeficienţilor $\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_m$, astfel încît suma patratelor diferenţei dintre y şi y_M să fie minimă.

Să notăm:

$$y_{Mi} = \alpha_0 + \sum_{j=1}^{m} \alpha_j x_{ji}\,;\ i = 1, 2, \ldots, n. \qquad (10-50)$$

Eroarea modelului este:

$$e_i = y_{Mi} - y_i = \alpha_0 + \sum_{j=1}^{m} \alpha_j x_{ji} - y_i\,;\ i = 1, 2, \ldots, n, \qquad (10-51)$$

Scanned with OKEN Scanner

iar funcţia erorii :

$$E = \sum_{i=1}^{n} e^2 = \sum_{i=1}^{n} \left(\alpha_0 + \sum_{j=1}^{m} \alpha_j x_{ji} - y_i \right)^2 . \qquad (10-52)$$

Valorile coeficienţilor $\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_m$ pentru care funcţia erorii este minimă se obţin din condiţiile de minimum, adică din relaţiile :

$$\frac{\partial E}{\partial \alpha_0} = \frac{\partial E}{\partial \alpha_1} = \frac{\partial E}{\partial \alpha_2} = \ldots = \frac{\partial E}{\partial \alpha_m} . \qquad (10-53)$$

Derivînd funcţia erorii în raport cu α_0, α_1, α_2, şi α_m şi egalînd cu zero expresiile derivatelor, se obţine următorul sistem de $m + 1$ ecuaţii cu $m + 1$ necunoscute pus sub formă matriceală :

$$\begin{pmatrix} n & \sum x_{1i} & \sum x_{2i} & \ldots & \sum x_{mi} \\ \sum x_{1i} & \sum x_{1i}^2 & \sum x_{1i}\, x_{2i} & \ldots & \sum x_{1i}\, x_{mi} \\ \sum x_{2i} & \sum x_{2i}\, x_{1i} & \sum x_{2i}^2 & \ldots & \sum x_{2i}\, x_{mi} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ \sum x_{mi} & \sum x_{mi}\, x_{1i} & \sum x_{mi}\, x_{2i} & \ldots & \sum x_{mi}^2 \end{pmatrix} \cdot \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \ldots \\ \alpha_m \end{pmatrix} = \begin{pmatrix} \sum y_i \\ \sum x_{1i}\, y_i \\ \sum x_{2i}\, y_i \\ \ldots \\ \sum x_{mi}\, y_i \end{pmatrix} . \qquad (10-54)$$

Pentru un proces cu o variabilă de ieşire y şi cu o variabilă de intrare x, ecuaţia matriceală se reduce la forma :

$$\begin{pmatrix} n & \sum x_i \\ \sum x_i & \sum x_i^2 \end{pmatrix} \cdot \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} = \begin{pmatrix} \sum y_i \\ \sum x_i y_i \end{pmatrix} \qquad (10-55)$$

din care se obţine soluţia următoare :

$$\alpha_0 = \frac{\sum y_i \sum x_i^2 - \sum x_i \sum x_i y_i}{n \sum x_i^2 - \left(\sum x_i \right)^2} \qquad (10-56)$$

$$\alpha_1 = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{n \sum x_i^2 - \left(\sum x_i \right)^2} .$$

Procedeul de mai sus este cunoscut şi sub denumirea de metoda celor mai mici pătrate [11].

Ex. 10—10. Fie x_i şi y_i din tabela 10—1 valorile variabilei de intrare şi respectiv ale variabilei de ieşire ale unui proces staţionar determinate la

Scanned with OKEN Scanner

zece momente de timp diferite pentru condiţii diferite. Se cere determinarea unui model liniar pe baza acestor date.

Tabela 10—1

Nr. crt.	x_i	y	x_i^2	$x_i y_i$
1	10	50	100	500
2	12	65	144	780
3	14	90	196	1260
4	16	120	256	1920
5	18	130	324	2340
6	20	145	400	2900
7	22	175	484	3850
8	24	180	576	4320
9	26	220	676	5720
10	28	220	784	6160
	$\sum_{i=1}^{10} x_i = 190$	$\sum_{i=1}^{10} y_i = 1395$	$\sum_{i=1}^{10} x_i^2 = 3940$	$\sum_{i=1}^{10} x_i y_i = 29\,750$

Calculele conduc la următoarele rezultate:

$$\alpha_0 = \frac{1395 \cdot 3940 - 190 \cdot 29750}{10 \cdot 3940 - 190^2} = -47{,}33,$$

$$\alpha_1 = \frac{10 \cdot 29750 - 190 \cdot 1395}{10 \cdot 3940 - 190^2} = 11{,}21.$$

Prin urmare ecuaţia modelului este:

$$y = 47{,}33 + 11{,}21x.$$

Calculele pot fi simplificate dacă se adoptă un nou sistem de referinţă cu originea în punctul de coordonate $\bar{y}, \bar{x}_1, \bar{x}_2, \ldots, \bar{x}_m$, unde $\bar{y}, \bar{x}_1, \bar{x}_2, \ldots, \bar{x}_m$ reprezintă valorile medii ale celor $m+1$ coordonate, adică

$$\bar{y} = \left(\sum_{i=1}^{n} y_i\right)/n; \quad \bar{x}_1 = \left(\sum_{i=1}^{n} x_{1i}\right)/n; \quad \bar{x}_m = \left(\sum_{i=1}^{n} x_{mi}\right)/n. \qquad (10-57)$$

Trecerea la noile coordonate se face după formula:

$$y_M^* = y_M - \bar{y}_M; \quad x_1^* = x_1 - \bar{x}; \quad \ldots, \quad x_m^* = x_m^- - \bar{x}_m, \qquad (10-58)$$

iar modelul de regresie este de forma:

$$y_M^* = \sum_{j=1}^{m} \alpha_j x_j. \qquad (10-59)$$

Construind funcţia erorii în noile coordonate, derivînd-o în raport cu parametrii $\alpha_1, \alpha_2, \ldots, \alpha_m$ şi egalînd cu zero aceste derivate obţinem urmă-

Scanned with OKEN Scanner

torul sistem de m ecuaţii cu m parametrii pus sub formă matriceală:

$$\begin{pmatrix} \sum_{i=1}^{n}(x_{1i}^*)^2 & \sum_{i=1}^{n} x_{1i}^* x_{2i}^* & \dots & \sum_{i=1}^{n} x_{1i}^* x_{mi}^* \\ \sum_{i=1}^{n} x_{2i}^* x_{1i}^* & \sum_{i=1}^{n}(x_{2i}^*)^2 & \dots & \sum_{i=1}^{n} x_{2i}^* x_{mi}^* \\ \dots & \dots & \dots & \dots \\ \sum_{i=1}^{n} x_{mi}^* x_{1i}^* & \sum_{i=1}^{n} x_{mi}^* x_{2i}^* & \dots & \sum_{i=1}^{n}(x_{mi}^*)^2 \end{pmatrix} \cdot \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \dots \\ \alpha_m \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{n} x_{1i}^* y_i^* \\ \sum_{i=1}^{n} x_{2i}^* y_i^* \\ \dots \\ \sum_{i=1}^{n} x_{mi}^* y_i^* \end{pmatrix}. \quad (10-60)$$

Din acest sistem se poate deduce expresia pentru fiecare din parametrii $\alpha_1, \alpha_2, \dots, \alpha_m$ sub forma:

$$\alpha = \mathbf{A}^{-1}\mathbf{B}, \quad (10-61)$$

unde **A** este matricea coeficienţilor, iar **B** — matricea din membrul drept al ecuaţiei (10—60).

Trecînd la vechiul sistem de coordonate, adică introducînd expresia (10—59) în expresia (10—58), se obţine:

$$y_M = \bar{y}_M + y_M^* = \bar{y}_M + \sum_{j=1}^{m} \alpha_j x_j, \quad (10-62)$$

de unde se deduce că:

$$\alpha_0 = \bar{y}_M - \sum_{j=1}^{n} \alpha_j \bar{x}_j. \quad (10-63)$$

Aproximarea funcţiilor de o singură variabilă printr-un polinom de forma:

$$y = a_0 + a_1 x + a_2 x^2 + \dots + a_m x^m, \quad (10-64)$$

utilizînd metoda celor mai mici pătrate, este prezentată în lucrarea noastră [11], la paginile 242—250, subcapitolul 4.2.3, unde este prezentat şi un program sursă în FORTRAN.

În legătură cu elaborarea modelelor de regresie se pune următoarea întrebare: în ce măsură modelul aproximează originalul sau care este gradul de precizie al modelului ca pe baza lui să se poată face o predicţie. Răspunsul se poate da pe baza analizei erorilor standard ale estimării şi a dispersiei datelor în jurul liniei de regresie.

Metode de corelaţie. Metodele de corelaţie fac posibilă estimarea funcţiei pondere şi a funcţiilor de frecvenţă pe baza datelor normale de exploatare determinate experimental.

Fie un proces aleator staţionar liniar caracterizat prin funcţia pondere $W(\theta)$. Variabila de ieşire a unui astfel de proces poate fi determinată din relaţia:

$$y(t) = \int_0^{\infty} W(\theta)\, x(t-\theta)\, d\theta, \quad (10-65)$$

unde $x(t-\theta)$ reprezintă variabila de intrare.

Scanned with OKEN Scanner

Dacă în această relaţie înlocuim pe t cu $t + \tau$ şi înmulţim ambele părţi ale egalităţii cu $x(t)$ obţinem:

$$y(t + \tau)\, x(t) = \int_0^\infty W(\theta)\, x(t - \tau - \theta)\, x(t)\, \mathrm{d}\theta. \qquad (10-66)$$

Media aritmetică pe o perioadă de timp $T \to \infty$ a ambilor termeni ai relaţiei (10—66) este:

$$\lim_{T\to\infty} \frac{1}{T} \int_0^T y(t + \tau)\, x(t)\, \mathrm{d}t = \lim_{T\to\infty} \frac{1}{T} \int_0^T \int_0^\infty W(\theta)\, x(t + \tau - \theta)\, \mathrm{d}\theta\, \mathrm{d}t. \qquad (10-67)$$

Schimbînd ordinea de integrare din partea dreaptă se obţine:

$$F_{xy}(\tau) = \int_0^\infty W(\theta)\, \mathrm{d}\theta \lim_{T\to\infty} \frac{1}{T} \int_0^T x(t + \tau - \theta) x(t)\, \mathrm{d}t = \int_0^\infty W(\theta) F_{xx}(\tau - \theta)\, \mathrm{d}\theta. \qquad (10-68)$$

Termenul din partea stîngă a relaţiei (10—68) reprezintă funcţia de intercorelaţie $F_{xy}(\tau)$, iar sub integrala din partea dreaptă a egalităţii se află funcţia de autocorelaţie $F_{xx}(\tau - \theta)$.

Determinarea funcţiei pondere $W(\theta)$ după funcţia de intercorelaţie $F_{xy}(\tau)$ şi după funcţia de autocorelaţie $F_{xx}(\tau - \theta)$, adică soluţionarea ecuaţiei de convoluţie, se poate realiza cu ajutorul calculatoarelor analogice sau numerice atît ca funcţii de timp cît şi ca funcţii de frecvenţă.

În domeniul timpului, această problemă se soluţionează relativ uşor dacă banda frecvenţelor semnalului $x(t)$ este mare în raport cu banda de trecere a procesului. În acest caz se poate face aproximaţia:

$$F_{xx}(\tau - \theta) = c\delta(\tau - \theta), \qquad (10-69)$$

unde $\delta(\tau - \theta)$ reprezintă impulsul Dirac, iar

$$c = \int_{-\infty}^{\infty} F_{xx}(\tau)\mathrm{d}\tau. \qquad (10-70)$$

Ţinînd cont de proprietăţile funcţiei δ se deduce:

$$F_{xy}(\tau) = cW(\tau),$$

$$W(\tau) = \frac{1}{c} F_{xy}(\tau). \qquad (10-71)$$

Rezultă deci că în acest caz funcţia pondere se deduce direct din funcţia de intercorelaţie.

Scanned with OKEN Scanner

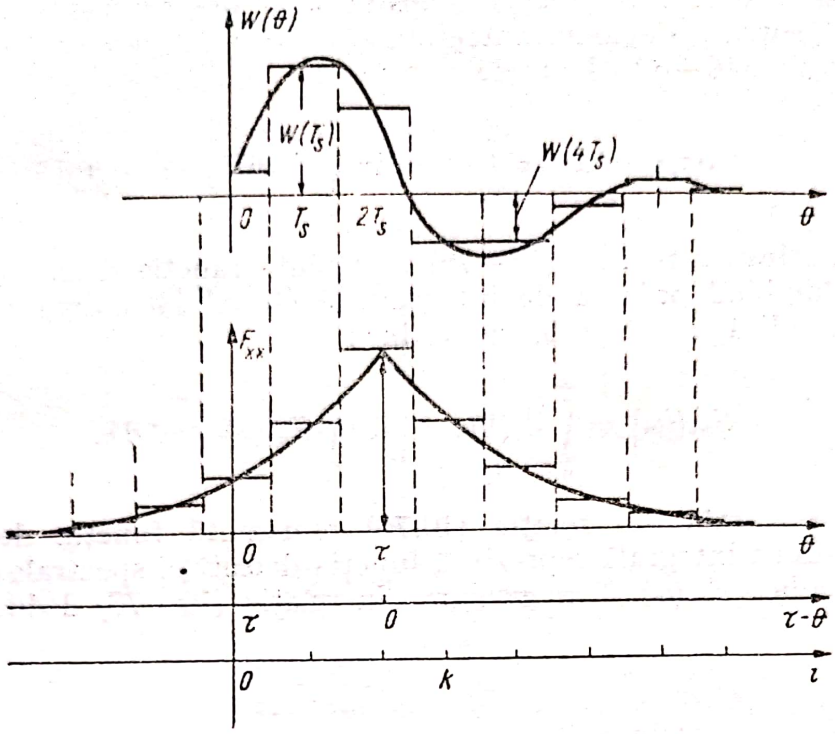


Fig. 10.34.

Dacă banda de frecvenţe a semnalului de intrare nu este largă atunci se recurge la aproximarea integralei (10.68) printr-o sumă de forma :

$$F_{xy}(\tau) = \int_0^{T_s/2} W(\theta)F_{xx}(\tau - \theta d\theta + \sum_{i=1}^{\infty} \int_{i-T_s/2}^{i+T_s/2} W(\tau)F_{xx}(\tau - \theta)d\theta, \tag{10—72}$$

sau ţinînd cont de notaţiile din figura 10.34 :

$$F_{xy}(kT_s) \simeq \frac{T_s}{2} W(0)F_{xx}(kT_s) + T_s \sum_{i=1}^{\infty} W(iT_s)F_{xx}((k-i)T_s). \tag{10—73}$$

În calitate de exemplu prezentăm ecuaţia (10.73) sub formă matriceală pentru patru valori discrete ale funcţiilor de corelaţie şi ale funcţiei pondere :

$$\begin{pmatrix} F_{xy}(0) \\ F_{xy}(T_s) \\ F_{xy}(2T_s) \\ T_{xy}(3T_s) \end{pmatrix} = T_s \begin{pmatrix} F_{xx}(0) & F_{xx}(-T_s) & F_{xx}(-2T_s) & F_{xx}(-3T_s) \\ F_{xx}(T_s) & F_{xx}(0) & F_{xx}(-T_s) & F_{xx}(-2T_s) \\ F_{xx}(2T_s) & F_{xx}(T_s) & F_{xx}(0) & F_{xx}(-T_s) \\ F_{xx}(3T_s) & F_{xx}(2T_s) & F_{xx}(T_s) & F_{xx}(0) \end{pmatrix} \begin{pmatrix} \frac{1}{2}W(0) \\ W(T_s) \\ W(2T_s) \\ W(3T_s) \end{pmatrix} \tag{10—74}$$

Din această ecuaţie, cunoscîndu-se două categorii de variabile (de exemplu funcţiile de intercorelaţie şi cele de autocorelaţie) se deduce cea de a treia categorie — funcţia pondere.

Scanned with OKEN Scanner

Funcţiile de corelaţie pot fi folosite şi la determinarea funcţiilor de frecvenţă. Pentru a demonstra aceasta să efectuăm transformarea Fourier asupra relaţiei (10—68) de unde obţinem:

$$\int_{-\infty}^{\infty} F_{xy}(\tau)e^{-i\omega\tau}d\tau = \int_{0}^{\infty} e^{-i\omega\tau}d\tau \int_{0}^{\infty} W(\theta)F_{xx}(\tau - \theta)d\theta. \qquad (10—75)$$

Partea stîngă a relaţiei (10—75) reprezintă funcţia densităţii spectrale $S_{xy}(i\omega)$. Schimbînd ordinea de integrare şi făcînd înlocuirea de variabilă $\tau - \theta = \lambda$, adică $\tau = x + \theta$, se obţine:

$$S_{xy}(i\omega) = \int_{0}^{\infty} W(\theta)e^{-i\omega\theta}\,d\theta \int_{0}^{\infty} F_{xx}(\lambda)e^{-i\omega\lambda}\,d\lambda. \qquad (10—76)$$

Prima integrală din relaţia (10.76) reprezintă funcţia de frecvenţă $Y(i\omega)$, iar a doua integrală reprezintă funcţia densităţii spectrale a variabilei de intrare, adică $S_{xx}(\omega)$. Prin urmare din relaţia (10—76) deducem:

$$Y(i\omega) = \frac{S_{xy}(i\omega)}{S_{xx}(\omega)}. \qquad (10—77)$$

Rezultă deci că funcţia de frecvenţă $Y(i\omega)$ se poate determina după funcţiile densităţii spectrale $S_{xy}(i\omega)$ şi $S_{xx}(\omega)$, care la rîndul lor se pot determina direct cu ajutorul unor analizoare spectrale sau după funcţiile de intercorelaţie, respectiv autocorelaţie prin transformarea Fourier.

Metoda prezentată poate fi extinsă şi la sisteme cu mai multe variabile de intrare, pe baza principiului superpoziţiei, valabil pentru sistemele liniare.

Metode iterative de gradient. Metodele iterative de estimare cu model acordabil oferă posibilitatea ca prin modificarea corespunzătoare a parametrilor modelului, aceştia să devină foarte apropiaţi (la limită — identici) cu parametrii procesului modelat. Acordarea modelului se poate face ca în figura 10.35.

Drept criteriu al erorii poate servi orice funcţie sau funcţională a erorii ca de exemplu funcţionala:

$$\mathbf{E} = \int_{t_1}^{t_2} \mathbf{e}^2(t,\ \mathbf{a},\ \alpha)dt, \qquad (10—78)$$

unde $\mathbf{e} = \mathbf{y} - \mathbf{y}_M$.

Direcţia acordării optime a modelului este indicată de semnul derivatei întîi a funcţionalei E în raport cu parametrii α. Pentru semnale continui, folosind expresia precedentă obţinem:

$$\nabla_\alpha \mathbf{E} = \frac{\partial \mathbf{E}}{\partial \alpha} = 2\int_{t_1}^{t_2} \mathbf{e}(t,\ \mathbf{a},\ \alpha)\ \frac{\partial \mathbf{e}}{\partial \alpha}\,dt, \qquad (10—79)$$

unde ∇_α este operatorul *nabla* referitor la variabila α.

Scanned with OKEN Scanner

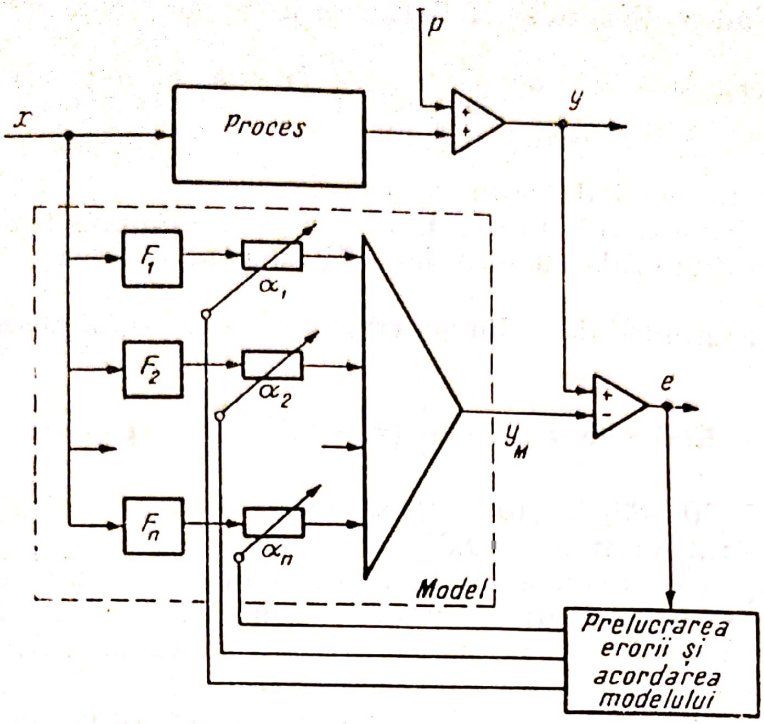


Fig. 10.35.

Dacă reuşim să determinăm analitic sau experimental valorile acestor derivate parţiale, atunci pentru estimarea celor mai bune valori ale parametrilor se pot folosi metode de gradient.

Să admitem că structura modelului este corect aleasă şi că dispunem de valorile eşantionate la intervalele de timp 1, 2, ..., k ale variabilelor de ieşire ale modelului şi procesului pentru aceeaşi valoare a variabilei de intrare comune.

Eroarea modelului adică $\mathbf{e} = \mathbf{y}_M - \mathbf{y}$ capătă valoarea minimă atunci cînd parametrii modelului $\alpha_1, \alpha_2, \ldots, \alpha_m$ devin identici cu parametrii procesului $a_1, a_2, \ldots, a_m$, adică pentru $\boldsymbol{\alpha} = \mathbf{a}$.

Pentru a minimiza eroarea în condiţiile cînd parametrii a ai procesului se modifică, este necesar a se determina în permanenţă sau periodic valorile derivatelor $\partial \mathbf{E}/\partial \boldsymbol{\alpha}$ şi adoptarea unei proceduri de acordare continuă sau discretă a parametrilor $\boldsymbol{\alpha}$.

Să presupunem că drept criteriu de apreciere a echivalenţei dintre model şi proces se adoptă o funcţie sau funcţională a erorii de forma :

$$\mathbf{E} = \mathbf{E}(\alpha_1, \alpha_2, \ldots, \alpha_m) = \mathbf{E}(\boldsymbol{\alpha}), \qquad (10-80)$$

unde $\boldsymbol{\alpha}$ sînt parametri acordabili ai modelului.

Funcţia erorii poate fi privită ca o hipersuprafaţă determinată de ecuaţia :

$$\mathbf{E}(\boldsymbol{\alpha}) - \mathbf{E}(\boldsymbol{\alpha} = \mathbf{a}) = 0. \qquad (10-81)$$

Dezvoltînd pe $\mathbf{E}(\alpha)$ în serie Taylor în jurul punctului $\alpha = \mathbf{a}$, se obţine:

$$\mathbf{E}(\alpha) = \mathbf{E}(\alpha)_{\alpha=\mathbf{a}} + (\alpha - \mathbf{a})\left(\frac{\partial \mathbf{E}}{\partial \alpha}\right)_{\alpha=\mathbf{a}} + \frac{1}{2}(\alpha - \mathbf{a})'\left(\frac{\partial^2 \mathbf{E}}{\partial \alpha' \partial \alpha}\right)_{\alpha=\mathbf{a}}(\alpha - \mathbf{a}) + \mathbf{R}, \tag{10—82}$$

unde $\mathbf{R}$ sînt termenii de ordin superior.

Întrucît în punctul $\alpha = \mathbf{a}$, $\mathbf{E}$ are valoare minimă, rezultă că $\partial \mathbf{E}/\partial \alpha$ pentru $\alpha = \mathbf{a}$ este egală cu zero, iar $\partial^2 \mathbf{E}/\partial \alpha^2$ pentru aceeaşi valoare a lui α este pozitivă.

Neglijînd termenii de ordin superior relaţia (10—82) poate fi transcrisă sub forma:

$$\mathbf{E}(\alpha) = \mathbf{E}(\alpha)_{\alpha=\mathbf{a}} + \frac{1}{2}(\alpha - \mathbf{a})\left(\frac{\partial^2 \mathbf{E}}{\partial \alpha^2}\right)_{\alpha=0}(\alpha - \mathbf{a}). \tag{10—83}$$

Înlocuind (10—83) în (10—81), se obţine ecuaţia unui hiperparaboloid care este ilustrat în figura 10.36.

Pentru aprecierea convergenţei algoritmilor de estimare a coeficienţilor α este necesar să se evidenţieze punctele, liniile sau suprafeţele pentru care funcţia $\mathbf{E}$ are aceeaşi valoare, adică $\mathbf{E} = \text{ct.}$ În cazul criteriului patratic aceste suprafeţe sînt elipsoizi cu mai multe axe.

Estimarea coeficienţilor modelului se poate realiza cu metode continue sau cu metode discontinue.

Dintre *metodele de acordare continuă* cea mai uzuală este metoda celei mai rapide coborîri conform căreia modificarea parametrilor modelului se face după o traectorie care asigură cea mai rapidă micşorare a erorii. Această traiectorie este perpendiculară pe linia de egală valoare a lui $\mathbf{E}$, adică pe linia care se obţine din ecuaţia:

$$\frac{\partial \mathbf{E}}{\partial \alpha} = \frac{\partial \mathbf{E}}{\partial \alpha_1} d\alpha_1 + \frac{\partial \mathbf{E}}{\partial \alpha_2} d\alpha_2 + \ldots = 0. \tag{10—84}$$

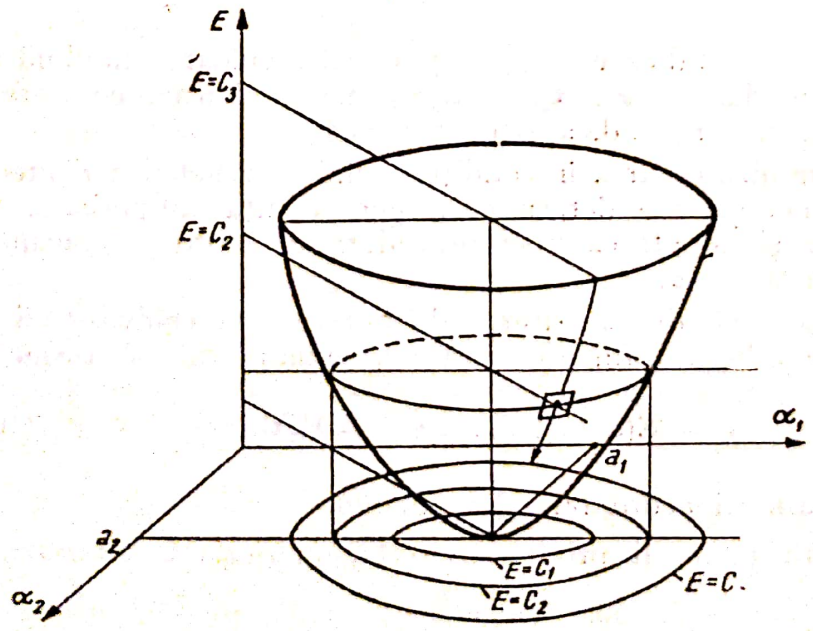


Fig. 10.36.

Scanned with OKEN Scanner

Ecuaţia tangentei la o astfel de linie în punctul $P(\alpha_{1p}, \alpha_{2p})$ dintr-un spaţiu bidimensional este:

$$\left(\frac{\partial E}{\partial \alpha_1}\right)_p (\alpha_1 - \alpha_{1p}) + \left(\frac{\partial E}{\partial \alpha_2}\right)_p (\alpha_2 - \alpha_{2p}) = 0. \qquad (10-85)$$

Vectorul transpus $\left(\frac{\partial E}{\partial \alpha_1}, \frac{\partial E}{\partial \alpha_2}\right)$ este perpendicular pe linia de egală valoare a lui **E**.

Conform metodei celei mai rapide coborîri acordarea parametrilor modelului se face după formula:

$$\begin{pmatrix} \dot{\alpha}_1 \\ \dot{\alpha}_2 \end{pmatrix} = -k \begin{pmatrix} \partial E/\partial \alpha_1 \\ \partial E/\partial \alpha_2 \end{pmatrix} = -k \frac{\partial E}{\partial \alpha}, \qquad (10-86)$$

în care k este o constantă de acordare care împreună cu derivatele parţiale determină viteza de modificare a parametrilor α.

În planul $\alpha_1 - \alpha_2$ traiectoria descrisă de ecuaţia (10—86) este perpendiculară pe liniile $E = ct$ (fig. 10.37) — traiectoria *1*). În realitate, o astfel de acordare ideală nu se poate face deoarece determinarea lui $\partial E/\partial \alpha$ şi deci schimbarea de direcţie nu se poate face instantaneu. De aceea traiectoria reală se abate de la cea ideală. Coeficientul k ce determină viteza de schimbare a parametrilor α trebuie să fie suficient de mare pentru a mări viteza şi micşora eroarea, dar suficient de mic pentru a nu duce la pierderea stabilităţii.

În practică se folosesc cu precădere metode discrete după care acordarea parametrilor se face la anumite intervale de timp, succesiv sau simultan.

Acordarea succesivă se face după următoarea procedură: se modifică unul din parametrii α, de exemplu α_1, pînă cînd $\partial E/\partial \alpha_1 = 0$, apoi se modifică parametrul următor pînă cînd $\partial E/\partial \alpha_2 = 0$ ş.a.m.d. se reia ciclul de la capăt pînă cînd se atinge zona cu cea mai mică valoare a lui E, adică zero.

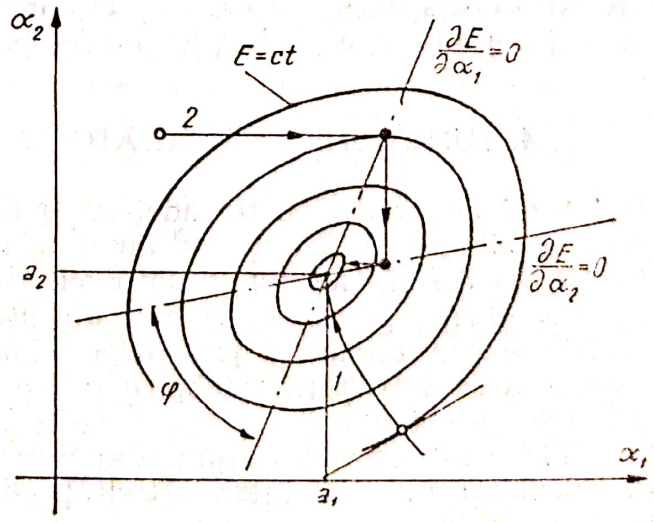


Fig. 10.37.

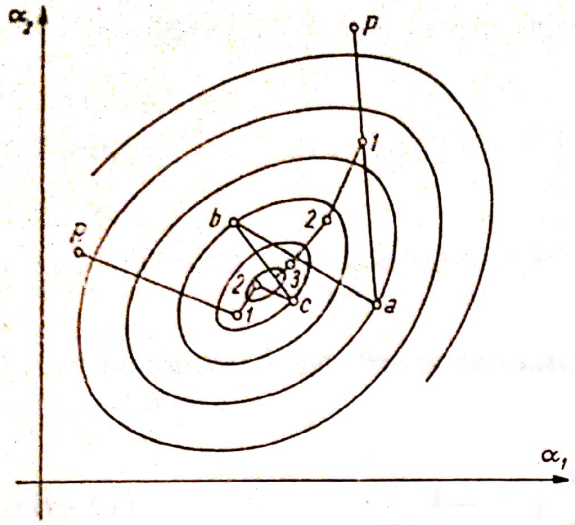


Fig. 10.38.

Din figura 10—37 — traiectoria 2, se poate deduce cu uşurinţă că numărul de cicluri va fi minim dacă liniile care constituie locul geometric al punctelor pentru care

$$\frac{\partial \mathbf{E}}{\partial \alpha_1} = 0\,; \quad \frac{\partial \mathbf{E}}{\partial \alpha_2} = 0$$

sînt octogonale. Cu cît unghiul dintre aceste linii va fi mai mic de 90° cu atît numărul de cicluri va fi mai mare.

Pentru *acordarea simultană* în paşi există mai multe metode dintre care menţionăm:

Metoda celei mai rapide coborîri care realizează acordarea parametrilor modelului după următoarea formulă:

$$\alpha(i+1) = \alpha(i) - k\frac{\partial \mathbf{E}(i)}{\partial \alpha(i)}\,; \quad k = \text{ct.} > 0 \qquad (10\text{—}87)$$

unde: $\alpha(i)$ este vectorul parametrilor modelului după pasul i; $\mathbf{E}(i)$ — valoarea funcţiei $\mathbf{E}$ după pasul i şi k — coeficientul de viteză de acordare.

Traiectoria şi viteza de apropiere de valorile optime ale parametrilor de acordare depind de punctul iniţial şi de valoarea constantei k.

Pentru valori mici ale lui k traiectoria va fi lină şi viteza — mică aşa cum este traiectoria P, 1, 2, 3 din figura 10.38, iar pentru valori foarte mari traiectoria va fi oscilantă iar viteza-mare, aşa cum este traiectoria P, a, b, c din aceeaşi figură.

Metoda gradientului iniţial se deosebeşte de precedenta prin aceea că deplasarea în direcţia gradientului iniţial se continuă pînă cînd se atinge valoarea minimă a lui E după care se organizează deplasarea pe o nouă direcţie pînă la atingerea unei noi valori minime a lui E ş.a.m.d. după cum se vede şi în figura 10.38. — traiectoria R, 1, 2.

Este uşor de observat că dacă liniile pentru care $\mathbf{E}$ = ct. ar fi nişte cercuri, acordarea optimă s-ar face într-un singur pas.

În literatura de specialitate există şi alte numeroase metode şi algoritmi de estimare (acordare) a coeficienţilor modelului.

10.4. FUNCŢIUNILE CALCULATOARELOR DE PROCES

Calculatorul de proces poate îndeplini o mare varietate de funcţiuni cu condiţia ca el să fie programat şi dotat în mod corespunzător. În cele ce urmează vom descrie succint următoarele categorii de funcţiuni:

— supravegherea proceselor şi echipamentelor tehnologice;
— conducerea automată a proceselor tehnologice;
— optimizarea activităţilor tehnice şi economice.

Descrierea este axată în principal pe aspectele care interesează pe utilizator, pentru a-l informa asupra resurselor deosebite pe care le poate oferi un calculator în conducerea activităţii de producţie.

Scanned with OKEN Scanner

Supravegherea proceselor şi echipamentelor

Una din cele mai uzuale funcţiuni ale calculatoarelor de proces constă în supravegherea desfăşurării proceselor tehnologice, care cuprinde următoarele forme:

— supravegherea aparatelor de măsurat şi analizoarelor;
— supravegherea stării echipamentului tehnologic;
— calculul unor indici sintetici;
— înregistrarea operativă a activităţii şi a evenimentelor importante;
— semnalizarea deficienţelor şi protecţia instalaţiilor tehnologice.

Funcţiunile de supraveghere constituie baza de plecare pentru exercitarea activităţii de conducere; de aceea se recomandă ca implementarea unui calculator de proces să înceapă tocmai cu exercitarea acestor funcţiuni. În cele ce urmează se vor prezenta pe scurt posibilităţile oferite de calculatorul de proces în acest domeniu.

Supravegherea aparatelor de măsurat şi analizoarelor. Calculatoarele de proces pot îndeplini mult mai bine decît personalul de exploatare funcţia de supraveghere a aparatelor de măsurat (manometre, debitmetre voltmetre, termometre etc.), a aparatelor de analiză a proprietăţilor fizico-chimice (densimetre, vîscozimetre etc.) şi a analizoarelor de compoziţie chimică (cromatografe, spectrografe ş.a.).

În cazul supravegherii automate, semnalele purtătoare de informaţie provenind de la aparatele de măsurat sau de la analizoare sînt mai întîi convertite, filtrate şi adaptate astfel încît să se încadreze în limitele admise la intrarea în calculator, apoi sînt convertite în cod numeric şi examinate de acesta.

Pe baza unor programe de exploatare, calculatorul efectuează mai întîi o serie de verificări pentru a constata dacă fiecare aparat de măsurat sau analizor funcţionează corect. Dacă acestea funcţionează corect, calculatorul trece la prelucrarea informaţiei primite. În caz contrar, printr-un mesaj corespunzător, indică aparatul defect, îl scoate din funcţiune şi, dacă este posibil, lucrează în continuare cu valori de substituţie.

Verificarea indicaţiilor aparatelor de măsurat şi analizoarelor se face de obicei pe baza examinării şi corelării informaţiei provenite de la mai multe surse, recurgîndu-se adesea şi la efectuarea unor calcule de bilanţ. De pildă, în regim staţionar, defectarea unuia din debitmetrele de pe un traseu poate fi depistată prin apariţia unor diferenţe în calculul bilanţului.

Informaţia furnizată de aparatele de măsurat şi analiză sînt prelucrate de către calculator pentru a elimina eventualele erori şi de a furniza rezultatul final al măsurării. Calculatorul de proces poate corecta deriva aparatelor de măsurat pe baza folosirii unei surse etalon, poate efectua corecţia de temperatură şi de presiune pentru un debitmetru diferenţial de gaze, corecţia pentru compensarea temperaturii capetelor libere ale unui termocuplu, corecţii pentru comportarea neideală a vaporilor ş.a. De asemenea poate executa operaţii de interpolare, extrapolare sau netezire a valorilor, operaţii de integrare a cromatogramelor, spectrogramelor sau a altor funcţii, operaţii de substituire a valorilor eronate şi multe altele.

Scanned with OKEN Scanner

În figura 10.39 este prezentată o procedură de citire, verificare și corecție a indicațiilor unui aparat de măsurat. Rutina începe prin preluarea semnalului standard purtător de informație, asocierea acestuia cu unitatea de măsură corespunzătoare (°C, kg/s, %etc.) și se termină prin afișarea variabilei măsurate, prin trecerea la procedura de alarmare sau prin trecerea la procedura de funcționare defectuoasă a aparatului respectiv.

Ciclul descris se repetă după fiecare interval de testare și pentru fiecare variabilă măsurată.

Supravegherea stării echipamentelor tehnologice. Calculatorul poate îndeplini un ansamblu de funcțiuni privind determinarea stării diverselor echipamente tehnologice precum și prevenirea sau împiedicarea atingerii unor stări inadmisibile. El poate determina și indica starea curentă a unor utilaje ca: pompe, compresoare, comutatoare, robinete, supape etc., pentru a furniza un tablou sintetic asupra stării instalației tehnologice.

Calculatoarele pot, de asemenea, preveni sau împiedica diversele combinații interzise între pozițiile anumitor robinete, comutatoare ș.a., pot determina și preveni regimurile de funcționare inadmisibile ale diverselor agregate eliminînd astfel pericolele care amenință securitatea personalului și instalațiilor, pierderile materiale ș.a.

Ex. 10—11. Pentru a evita accidentele ce le-ar putea provoca pătrunderea necontrolată a combustibilului în focarul unui cuptor tubular, pe conducta de combustibil, înainte de robinetul de reglare se montează un robinet de închidere. Acționarea (închiderea/deschiderea) acestuia poate fi realizată, sau cel puțin condiționată, de către un calculator de proces. În acest scop, pe baza supravegherii aparatelor de măsurat, calculatorul poate obține informația necesară rezolvării unei ecuații logice prin care stabilește dacă robinetul de combustibil trebuie să fie deschis sau închis.

O astfel de ecuație poate fi de forma:

$$V = E \cap P \cap G \cap \overline{C}_0 \cap A_p \cap C_\varepsilon \cap \overline{C}_p, \qquad (10-88)$$

unde V, E, P, G, C_0, A_p, C_ε și C_p sînt variabile binare ale căror valori „adevărate" au următoarele semnificații:

$V = 1$ — robinetul în poziția deschis;
$E = 1$ — cuptorul este alimentat cu energie;
$P = 1$ — cuptorul este alimentat cu produs;
$G = 1$ — cuptorul dispune de combustibil;
$C_0 = 1$ — s-a dat comanda de oprire a cuptorului;
$A_p = 1$ — arzătoarele pilot sînt aprinse;
$C_\varepsilon = 1$ — s-a dat comanda de deschidere a robinetului;
$C_p = 1$ — s-a dat comanda de purjare cu abur a cuptorului.

Prin urmare, calculatorul comandă sau condiționează deschiderea robinetului de combustibil în funcție de condițiile existente, adică de valorile variabilelor logice care pot fi determinate prin intermediul unor traductoare binare.

Robinetul va putea fi deschis dacă: $E = 1$, $P = 1$, $G = 1$, $C_0 = 0$, $A_p = 1$, $C_\varepsilon = 1$ și $C_p = 0$. În caz contrar el va fi închis.

Calculatoarele pot determina și starea de uzură a echipamentului tehnologic sau alte stări anormale ale acestuia. De exemplu, pe baza măsurării pierderii de presiune pe care o are un agent termic la traversarea

Scanned with OKEN Scanner

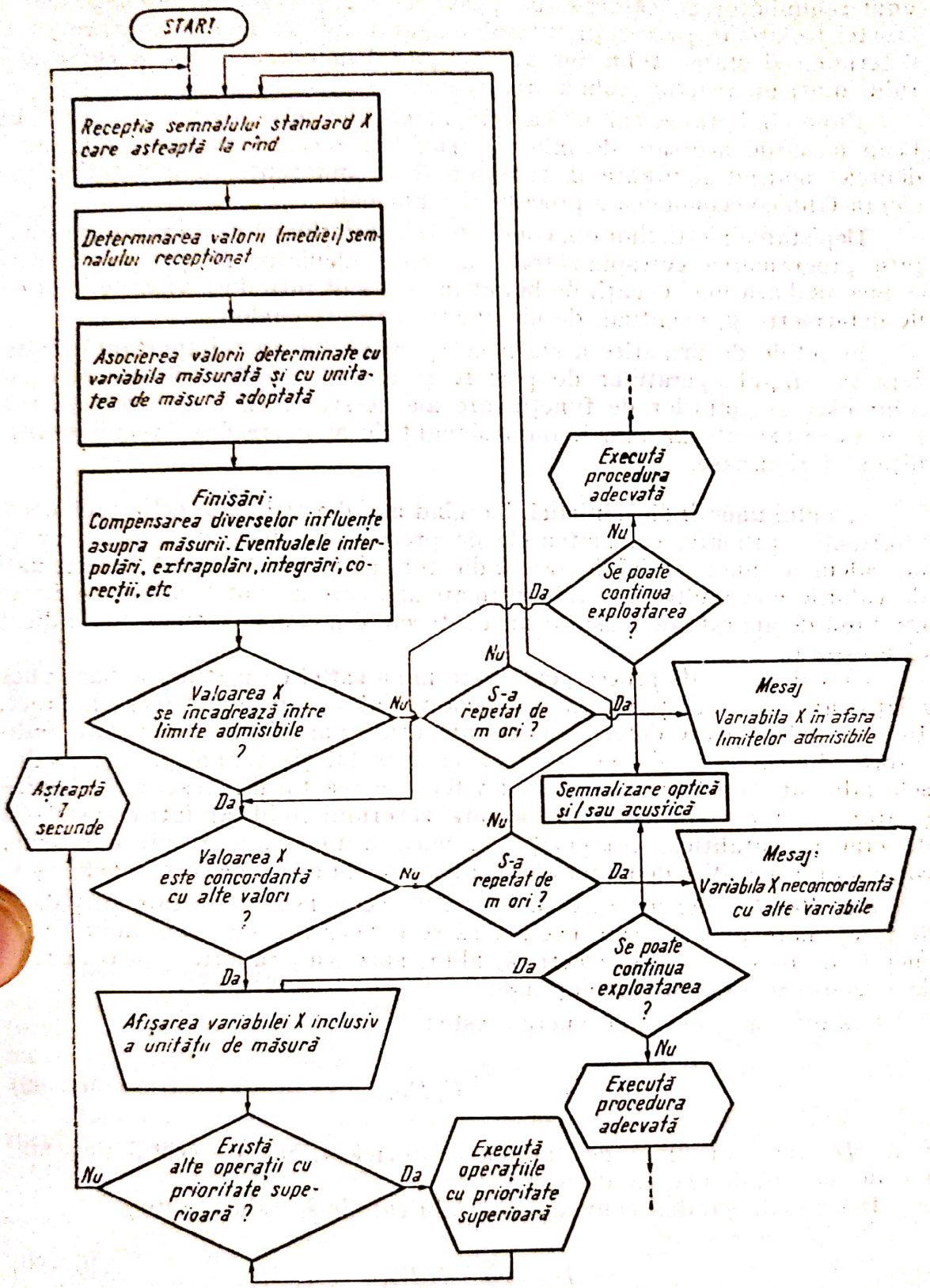


Fig. 10.39.

Scanned with OKEN Scanner

Dana este acasă cu iubitul ei.
Dana este acasă cu Mian.

unui schimbător de căldură, se poate determina gradul de depunere a crustei izolatoare pe pereţii tuburilor aparatului. În mod similar, se poate determina şi starea tuburilor unui cuptor tubular, activitatea catalizatorului dintr-un reactor, colmatarea filtrelor ş.a.

Cunoscînd starea curentă a echipamentului tehnologic se pot lua din timp măsurile necesare de înlocuire sau remediere, prevenind astfel accidentele, mărind siguranţa în funcţionare şi anduranţa, îmbunătăţind eficienţa tehnico-economică a procesului tehnologic.

Depistarea pierderilor din conducte şi instalaţii este de asemenea posibilă prin programarea corespunzătoare a unui calculator de proces. Aceasta se face pe baza unei ecuaţii de bilanţ în care sînt introduse valorile indicate de debitmetre şi, eventual, de nivelmetrele rezervoarelor.

Funcţiile de urmărire a stării echipamentului au o importanţă deosebită în timpul operaţiilor de pornire şi oprire a instalaţiilor şi în cazul schimbării regimurilor de funcţionare ale acestora. În aceste situaţii trebuie să se execute un număr impresionant de manevre după reguli şi condiţionări riguroase.

Calculul unor indici sintetici. Mergînd mai departe în direcţia prelucrării informaţiei primare, calculatoarele de proces pot îndeplini diverse funcţii de calcul al unor variabile sau indicatori sintetici care sînt determinaţi de valorile mai multor variabile primare sau care nu pot fi măsurate direct din lipsă de aparate de măsurat adecvate sau din cauza costului prea ridicat al acestora.

Un calculator de proces poate determina astfel de mărimi pe baza unor relaţii care leagă aceste mărimi de unele variabile care se pot măsura direct. În acest fel se poate determina randamentul unor maşini şi agregate, indicatori tehnico-economici ce caracterizează instalaţia tehnologică şi produsele fabricate, cum ar fi de pildă: viteza de reacţie din reactoare chimice şi atomice, viteza de circulaţie a catalizatorului fluidizat într-o instalaţie de cracare catalitică, compoziţia chimică a unor amestecuri complexe, consumurile specifice de materiale şi/sau energie, preţuri de cost, beneficii ş.a.

Ex. 10—12. Pentru analiza eficienţei economice a oricărei instalaţii de fracţionare primară este necesar să se cunoască costul consumurilor de energie de diverse forme (electrică, abur, gaze ş.a.) raportată la o unitate de măsură de produs convenţional.

Costul consumului de energie este:

$$C_e = \sum_{i=1}^{n} C_{ei} D_{ei}, \qquad (10-89)$$

unde D_{ei} este consumul de energie de forma i, iar C_{ei} costul unei unităţi de măsură de energie de forma i.

Debitul de produs convenţional fabricat de instalaţie este:

$$D_{pc} = \sum_{j=1}^{m} C_{pj} D_{pj}, \qquad (10-90)$$

unde D_{pj} este debitul de produs din sortimentul j, fabricat de instalaţie, iar C_{pj} un factor de conversie care exprimă raportul dintre valoarea

Scanned with OKEN Scanner

unității de măsură din sortimentul j și valoarea unității de măsură de produs convențional (de exemplu: 1 tonă de benzină convențională valorează cît 4 tone de motorină, deci $C = 0{,}25$).

Costul consumului specific instantaneu de energie este:

$$C_s = C_e/D_{pc} = \sum_{i=1}^{n} C_{ei}D_{ei} \Big/ \sum_{j=1}^{m} C_{pj}D_{pj}, \qquad (10-91)$$

iar costul consumului specific mediu de energie raportat la o perioadă de timp $t = N\Delta t$ este:

$$C_{sm} = \sum_{k=1}^{N}\left(\sum_{i=1}^{n} C_{ei}D_{ei}\right)_k \Delta t_k \Big/ \sum_{k=1}^{N}\left(\sum_{j=1}^{m} C_{pj}D_{pj}\right)_k \Delta t_k. \qquad (10-92)$$

Un calculator de proces, programat corespunzător, poate furniza personalului de conducere (la cerere sau, dacă este necesar, chiar în timp real) toate cele patru mărimi: C_e, D_{pc}, C_s și C_{sm}.

Efectuînd diferite calcule și prezentînd personalului de conducere rezultatele acestora, calculatorul „condensează" o mare cantitate de informație primară „brută" într-un număr mic de variabile (indicatori) fundamentale și ajută acest personal la luarea unor decizii, chiar dacă nu îi indică ce anume decizie să ia.

Dispunînd de calculator, personalul de conducere este scutit de manipularea și prelucrarea informației primare pe care de altfel, în cazul proceselor tehnologice complexe, nici n-ar putea să o facă în timp util. El primește direct rezultatul acestei activități și își poate concentra atenția asupra funcțiilor de decizie.

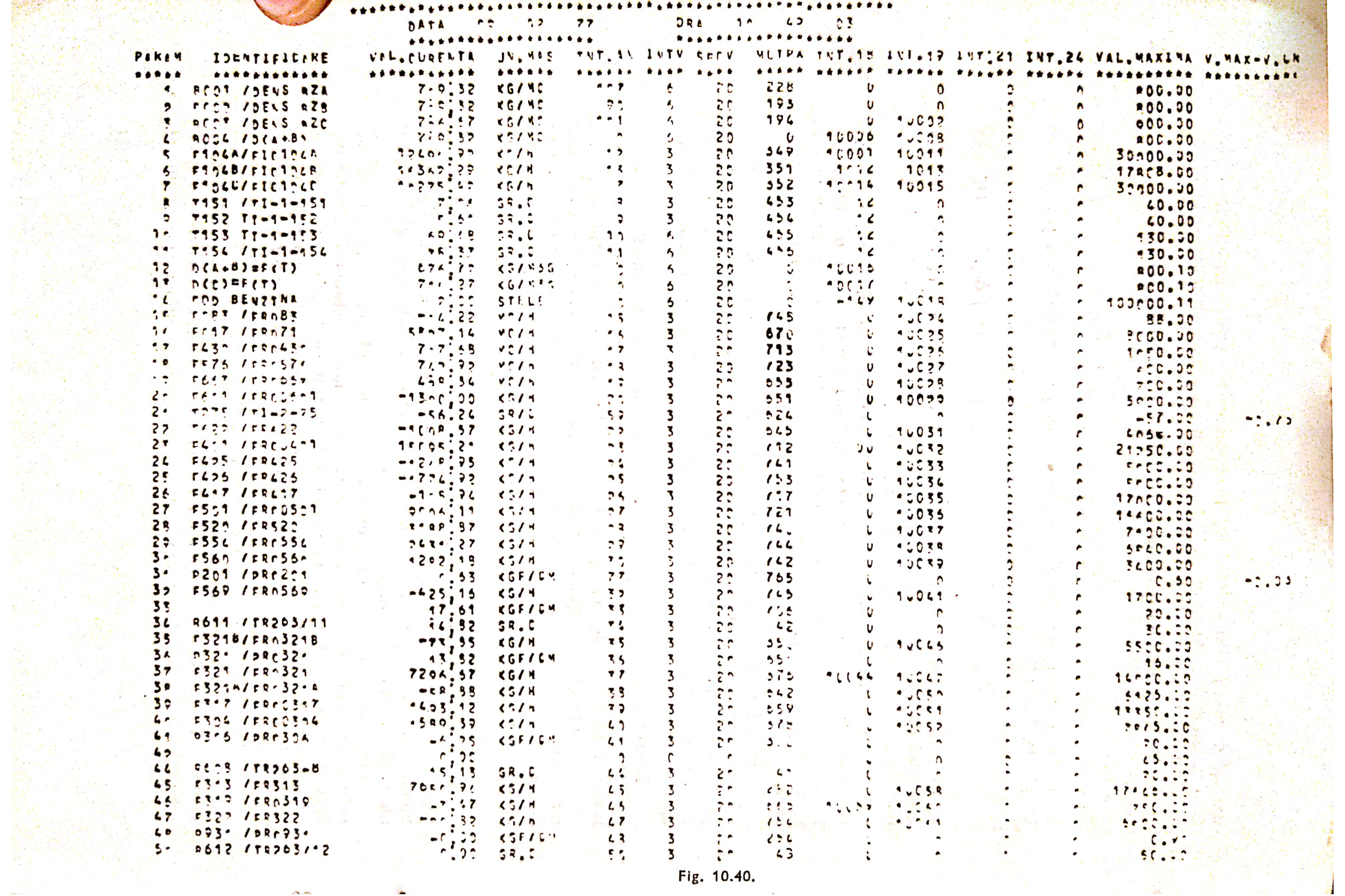
Deosebit de eficace este utilizarea calculatorului în determinarea indicatorilor economici ai activității de producție, reflectați în ultimă instanță în prețul de cost și în beneficii. Determinarea acestor indicatori sintetici servește nu numai conducerii proceselor tehnologice ci și unei mai bune cunoașteri a proceselor respective cu scopul măririi eficienței acestora, calculatorul fiind un „măsurător" al eficienței și profitului, un „profitmetru".

Înregistrarea operativă a activităților și a evenimentelor importante. Pe baza supravegherii procesului și echipamentului tehnologic, calculatorul de proces poate îndeplini funcția de evidență operativă a unor evenimente importante precum și funcția de raportare a activității globale pe o anumită perioadă de timp.

Programat în mod corespunzător, calculatorul poate prezenta personalului de conducere sub formă scrisă, pe formulare standard sau pe fișe speciale, valorile principalelor variabile măsurate ca și pe cele determinate prin calcul, redactînd adevărate „jurnale de bord" ale instalației supravegheate. Un exemplu de cap de tabel redactat de calculator este prezentat în figura 10.40.

Pe baza supravegherii aparaturii de măsurat și a stării echipamentului tehnologic calculatorul poate prezenta o imagine instantanee a desfășurării procesului tehnologic la anumite intervale de timp, precum și efectul integrat al stărilor instantanee care interesează nu numai personalul de

Scanned with OKEN Scanner



exploatare a instalației tehnologice, ci și organele de decizie de la nivelele superioare.

Evidența pe calculator a evoluției parametrilor importanți ai instalației tehnologice cum ar fi, de exemplu, temperatura și presiunea într-un reactor chimic, debitul de materie primă dintr-un cuptor tubular, consumul de energie ș.a. servește nu numai scopurilor tehnologice, ci rămîne ca document pentru o analiză a cauzelor anumitor accidente sau deranjamente, durata opririlor instalației, funcționarea sub sau peste parametrii proiectați, timpi morți globali sau raportați la un anumit colectiv și la interval de timp ș.a.

De asemenea, evidența pe calculator a stocurilor, a consumurilor de materii prime și auxiliare, a consumului de energie, a producției globale și pe sortimente, a productivității muncii pe instalație sau pe un colectiv, a calității produselor fabricate, a prețului de cost, a beneficiilor și a altor indicatori tehnico-economici este mult mai eficace și veridică decît cea clasică, ținută de o echipă numeroasă de operatori și funcționari cu inerentele lor posibilități de a greși. Această evidență este folosită nu numai ca document tehnic ci și ca document contabil în operațiile de planificare și contabilizare a producției.

Pe baza unei astfel de evidențe calculatorul este în măsură să prezinte, periodic sau la cerere, situația globală sub forma cerută de utilizator și să redacteze adevărate „protocoale" pe o anumită perioadă de timp, să redacteze și să emită buletine de cantitate și calitate a produselor fabricate, să calculeze și să redacteze statele de plată pentru angajații instalației respective pe baza pontajului acestora, precum și alte activități specifice unor calculatoare universale.

Mai mult decît atît, pe baza unui program corespunzător, calculatorul poate prezenta solicitantului diagnosticul (cauza) și prognoza în ceea ce privește desfășurarea unui anumit eveniment cum ar fi, de exemplu, tendința de variație a prețului de cost pe baza extrapolării ultimelor date, calculatorul fiind un instrument deosebit de eficace în sistemul informațional global al unei întreprinderi.

O altă aplicație deosebit de interesantă este efectuarea de către calculator a bilanțurilor de materiale și/sau energie pe diverse instalații și pe întreprindere cu scopul depistării pierderilor și ameliorării condițiilor de exploatare a instalațiilor tehnologice.

Semnalizarea deficiențelor și protecția instalațiilor. Un calculator de proces poate îndeplini și funcții de semnalizare „alarmare" a unor stări anormale sau periculoase, precum și funcții de protecție automată a instalației și personalului în caz de pericol.

Dacă este programat și conectat la proces în mod corespunzător, calculatorul poate avertiza printr-un mesaj scris și un semnal sonor și/sau optic dacă vreuna din variabilele procesului sau vreunul din parametrii de stare ai instalației tehnologice au ieșit din limitele admisibile pentru a determina personalul de exploatare să ia măsurile de rigoare sau să intervină direct în exercitarea acestora.

În figura 10.41 este prezentată o procedură de semnalizare (alarmare) a depășirilor de către principalele variabile ale unui proces, a limitelor admisibile superioare și inferioare.

Scanned with OKEN Scanner

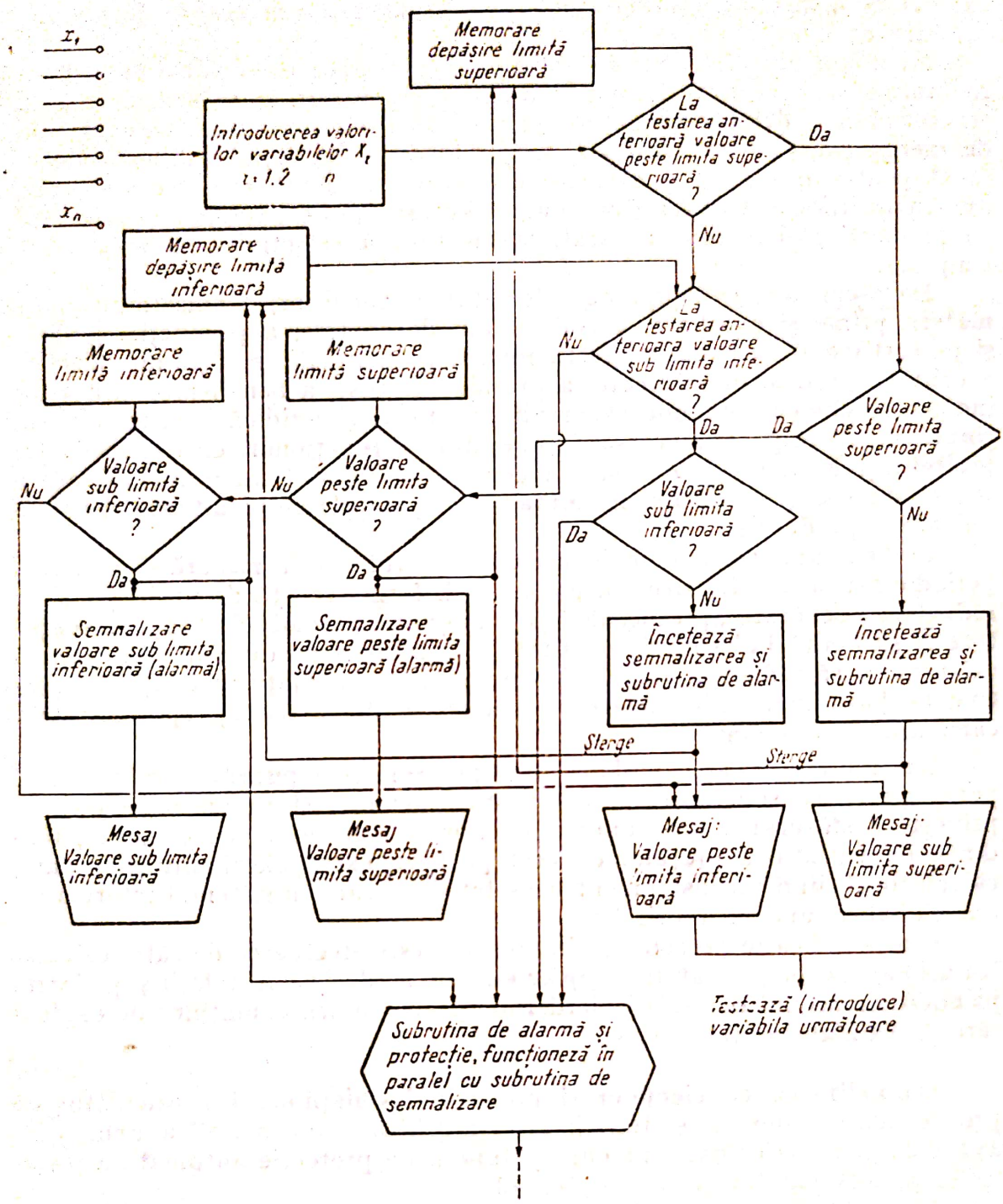


Fig. 10.41.

Conform acestei proceduri, variabilele procesului sînt testate ciclic. Se verifică mai întîi dacă variabila testată a depășit limita superioară sau inferioară la testarea precedentă. Dacă o astfel de depășire a avut loc, se verifică dacă variabila respectivă se încadrează în limitele admisibile. În caz afirmativ se încetează semnalizarea de alarmare și execuția subrutinei de alarmă, se semnalizează reîncadrarea variabilei între limitele admise și

Scanned with OKEN Scanner

se trece la testarea următoarei variabile. În caz contrar rămîne în acţiune subrutina de alarmă.

După verificarea rezultatului testării anterioare se testează valoarea curentă a variabilei x prin comparare cu limitele superioare şi, respectiv inferioară admise. Dacă variabila testată depăşeşte vreuna din limitele admise acesta atrage după sine semnalizarea de alarmare şi intrarea în acţiune a subrutinei de alarmă. Depăşirea constatată se memorează pentru a fi luată în seamă la testarea următoare. Dacă variabila se încadrează între limitele admise calculatorul indică aceasta printr-un mesaj corespunzător, prin lămpi colorate sau în scris.

Marea varietate de situaţii anormale posibile în desfăşurarea unui proces impune şi o mare varietate de măsuri de protecţie. Dintre acestea se poate menţiona oprirea integrală a instalaţiei după anumite reguli, în funcţie de situaţia creată, oprirea unei părţi a instalaţiei, adică partea afectată de pericol, funcţionarea în regim special a întregii instalaţii sau numai a părţii afectate de pericol ş.a. Aşa, de exemplu, dacă presiunea pe conducta de refulare a unei pompe a scăzut brusc ca efect al spargerii conductei respective, iar bilanţul materiei nu se mai încheie, calculatorul poate comanda oprirea pompei în cauză, închiderea robinetelor corespunzătoare precum şi alte măsuri în funcţie de gravitatea accidentului respectiv.

Pentru a preveni accidentele şi avariile, calculatorul poate fi programat să semnaleze tendinţele nedorite sau periculoase înainte de a atinge limitele periculoase. De pildă, dacă în ultimele zile căderea de presiune pe serpentina unui cuptor tubular şi temperatura acesteia au crescut progresiv, calculatorul poate avertiza personalul de exploatare asupra pericolului spargerii serpentinei din cauza creşterii presiunii şi temperaturii acesteia ca rezultat al depunerii crustei termoizolante.

Conducerea automată a proceselor

Conducerea nemijlocită „on line" a proceselor tehnologice cu ajutorul calculatoarelor constituie o treaptă superioară în evoluţia utilizării acestor mijloace moderne de automatizare. Diversele aplicaţii ale calculatoarelor în această direcţie pot fi încadrate în următoarele două domenii: *conducerea secvenţională a operaţiilor* şi *comanda/reglarea automată multiplă*.

În următoarele trei paragrafe vom prezenta aspectele de bază care interesează îndeosebi pe utilizator.

Conducerea secvenţială a operaţiilor. Sînt numeroase situaţiile în care fazele sau operaţiile unui proces sînt secvenţiale, unde înainte de a trece la faza sau operaţia următoare este necesar să se analizeze rezultatul fazelor anterioare şi în funcţie de rezultat să se aleagă o variantă din numeroase alternative posibile.

Efectuarea unor astfel de funcţii se face pe baza unor modele procedurale sau procedural-calculatorii care pot fi incorporate în programe adecvate şi introduse în calculator în vederea automatizării funcţiilor respective. Un exemplu de astfel de model este prezentat în figura 10.20.

Un caz caracteristic de conducere secvenţială îl constituie pornirea şi oprirea unei instalaţii tehnologice, de pildă o instalaţie de distilaţie at-

Scanned with OKEN Scanner

mosferică a petrolului, unde trebuie coordonate sute sau mii de acțiuni de la începerea pornirii sau opririi pînă la terminarea ei.

Aici, înainte de a începe o acțiune se examinează mai întîi aparatura de măsurat și control pentru a constata rezultatul acțiunilor anterioare și în funcție de acest rezultat se alege o alternativă din numeroasele căi posibile de urmat pe baza unei logici dinainte stabilite. După efectuarea fiecărui pas din suita logică a operațiilor, ciclul descris se repetă pînă cînd uriașul labirint al alternativelor a fost parcurs în întregime.

Evident că utilizarea unui calculator exclusiv pentru acest scop nu poate fi justificată, deoarece astfel de operații se efectuează relativ rar. Poate fi însă utilizat un calculator care în mod normal îndeplinește alte funcții (supraveghere, reglare etc.), și care programat corespunzător pentru pornire/oprire asigură realizarea acestor operații în timp mai scurt și în deplină securitate.

Un alt exemplu de conducere secvențială îl constituie executarea operațiilor de alarmare și protecție a instalațiilor în caz de avarie sau funcționare anormală. Trecerea de la un regim de funcționare normală la altul, ca și trecerea de la un regim anormal la altul sau la o oprire forțată a unei instalații se face tot secvențial după un program în care trebuie prevăzute toate alternativele posibile.

Conducerea unor procese tehnologice discontinui sau ciclice se face de asemenea secvențial. Ca exemplu putem da instalațiile tehnologice cu mai multe reactoare la care regenerarea catalizatorului trebuie să se facă la intervale de timp relativ scurte și unde pentru asigurarea continuității procesului reactoarele urmează să funcționeze succesiv în regim de lucru și respectiv în regim de regenerare prin conectarea lor corespunzătoare la rețeaua de conducte a instalației.

În figura 10.42 este prezentată schema unei baterii de trei reactoare și procedura de exploatare a acestora.

Înainte de a trece la faza de reacție, fiecare reactor este testat pentru a verifica dacă este pregătit pentru această fază. Trecerea este condiționată de rezultatul următoarei funcții logice:

$$R_k = R_{gk} \cap \bar{V}_{ik} \cap \bar{V}_{ek} \cap v_{ik} \cap v_{ek}; \quad k = 1, 2, 3, \qquad (10-93)$$

unde simbolurile au următoarele semnificație:

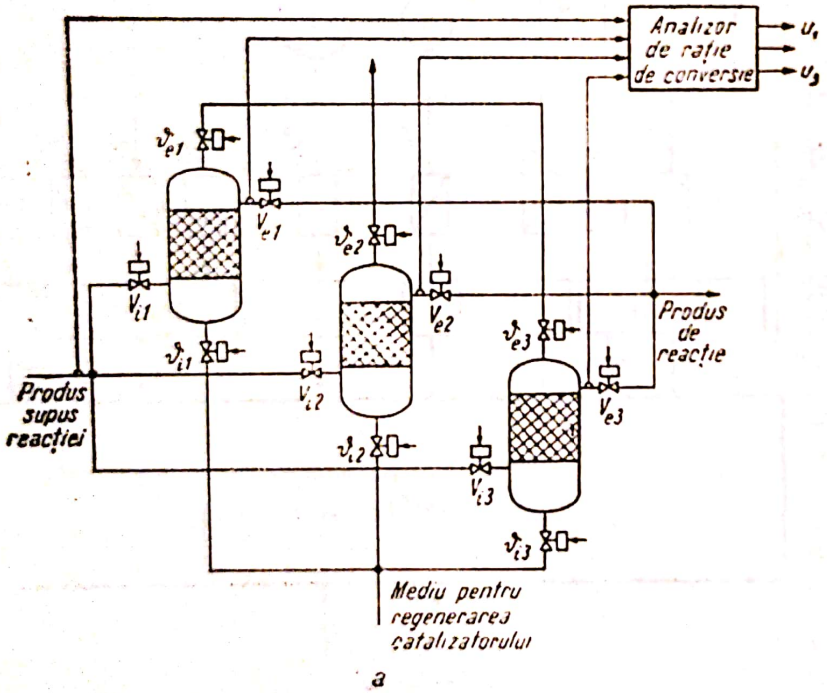
$R_k = 1$ — reactorul k este pregătit pentru faza de reacție;
$R_{gk} = 1$ — regenerarea catalizatorului a avut loc;
$V_{ik} = 1$ — robinetul de intrare produs este închis;
$V_{ek} = 1$ — robinetul de ieșire produs este închis;
$v_{ik} = 1$ — robinetul de intrare abur-regenerare este închis;
$v_{ek} = 1$ — robinetul de ieșire abur este închis.

Faza de reacție durează pînă în momentul cînd U_k — rația de conversie — devine mai mică decît U_{pk} — rația de conversie prescrisă dar nu mai mult de T minute pentru a nu uza în mod excesiv catalizatorul din reactorul i.

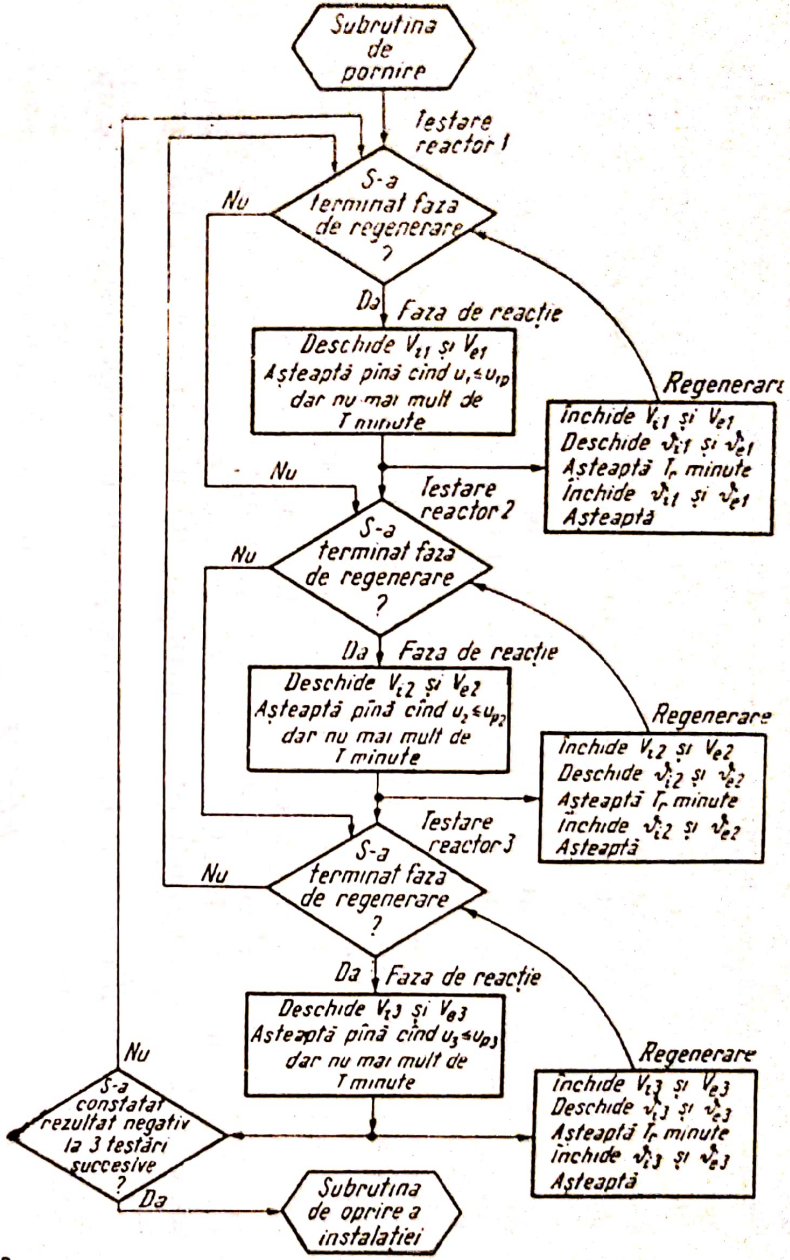
Faza de regenerare începe prin închiderea robinetelor de produs și deschiderea robinetelor de abur pentru regenerare și durează timp de T_r minute după care se închid și robinetele de abur.

Dacă după trei testări succesive nici unul din reactoare nu este apt pentru faza de reacție instalația se oprește.

Scanned with OKEN Scanner


Analizor de rație de conversie
u_1
u_3
V_{e1}
V_{i1}
V_{e2}
V_{i2}
V_{e3}
V_{i3}
Produs supus reacției
Produs de reacție
Mediu pentru regenerarea catalizatorului

a


Subrutina de pornire
Testare reactor 1
S-a terminat faza de regenerare ?
Nu
Da
Faza de reacție
Deschide V_{i1} și V_{e1}
Așteaptă pînă cînd $u_1 \leq u_{1p}$ dar nu mai mult de T minute
Regenerare
Închide V_{i1} și V_{e1}
Deschide V_{i1} și V_{e1}
Așteaptă T_r minute
Închide V_{i1} și V_{e1}
Așteaptă
Testare reactor 2
S-a terminat faza de regenerare ?
Deschide V_{i2} și V_{e2}
Așteaptă pînă cînd $u_2 \leq u_{p2}$ dar nu mai mult de T minute
Închide V_{i2} și V_{e2}
Deschide V_{i2} și V_{e2}
Așteaptă T_r minute
Închide V_{i2} și V_{e2}
Așteaptă
Testare reactor 3
S-a terminat faza de regenerare ?
Deschide V_{i3} și V_{e3}
Așteaptă pînă cînd $u_3 \leq u_{p3}$ dar nu mai mult de T minute
Închide V_{i3} și V_{e3}
Deschide V_{i3} și V_{e3}
Așteaptă T_r minute
Închide V_{i3} și V_{e3}
Așteaptă
S-a constatat rezultat negativ la 3 testări succesive ?
Subrutina de oprire a instalației

b

Fig. 10.42.

Comanda și reglarea automată multiplă. În acest domeniu calculatoarele de proces pot fi utilizate în două moduri principial deosebite și anume:

— calculatorul se suprapune peste sistemele clasice de reglare automată, completînd astfel instalația de automatizare;

— calculatorul înlocuiește o bună parte a sistemelor de comandă și reglare, îndeplinind aceste funcții în mod nemijlocit, realizînd ceea ce se numește comandă numerică directă, simbolizată prin *DDC* de la „*D*igital *D*irect Control".

În primul caz, calculatorul este utilizat în principal la calculul valorilor de referință pentru regulatoare și la transmiterea acestor valori regulatoarelor.

În al doilea caz, calculatorul determină — pe baza analizei desfășurării procesului tehnologic — acțiunile ce trebuie întreprinse pentru a asigura desfășurarea normală a acestuia și le și execută nemijlocit fără a mai utiliza în acest scop regulatoare ca elemente intermediare, ci acționînd direct asupra elementelor de execuție din instalația tehnologică.

Pentru a înțelege particularitățile și avantajele *DDC*, să analizăm instalația din figura 10.43, alcătuită din sisteme de reglare automată analogice individuale. Caracteristic pentru această instalație sînt următoarele aspecte:

— pentru fiecare variabilă se folosește cîte un *SRA*, respectiv cîte un regulator;

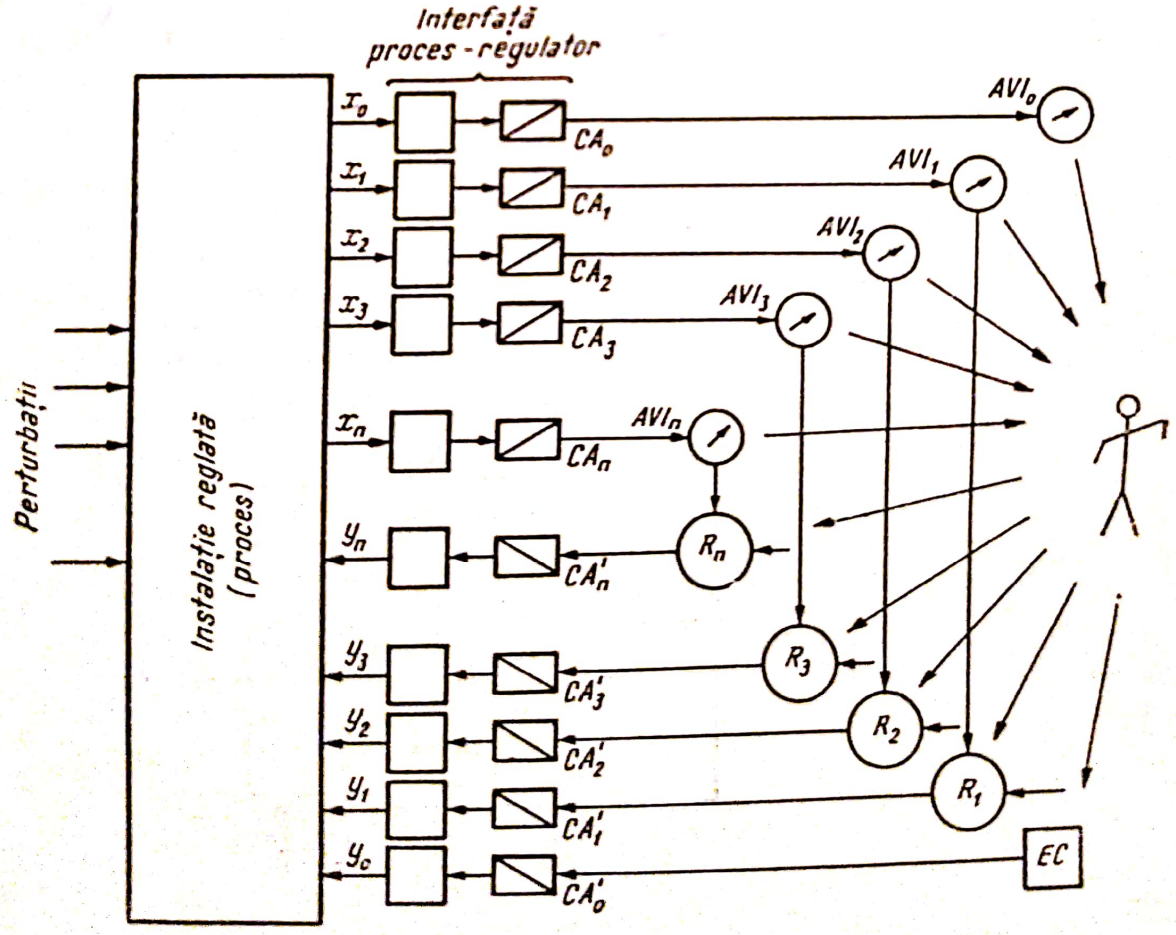


Fig. 10.43.

Scanned with OKEN Scanner

— fiecare regulator este conectat permanent la unul şi acelaşi *SRA* ;
— regulatoarele analogice utilizate nu pot fi programate pentru alte funcţii decît cele pentru care au fost construite ;
— vizualizarea informaţiei privind : variabila reglată, abaterea, variabila de referinţă ş.a. se face de regulă cu aparate individuale.

Este evidentă o funcţionare în paralel a tuturor *SRA* şi — condiţionat de aceasta — o repetare a echipamentului de reglare, în speţă a regulatoarelor, a convertoarelor-adaptoare, a aparaturii de vizualizare, a traductoarelor şi a elementelor de execuţie, deci o individualizare extremă a acestui echipament.

La baza conceptului de *DDC* stă ideea înlocuirii, în măsura în care este posibil, a echipamentului de comandă şi/sau reglare analogică (regulatoare, convertoare ş.a.) şi a celui de vizualizare a informaţiei, care conform conceptului clasic lucrau în paralel, cu elemente componente ale unui calculator de proces care se conectează şi lucrează succesiv, deci în serie, în fiecare circuit informaţional (fig. 10.44).

Conform acestui concept modern, elemente ale unui calculator de proces sau întreg calculatorul preia funcţiile de centralizare şi prelucrare a informaţiei sub formă numerică, ca şi funcţia de vizualizare a acesteia, programul de prelucrare şi toate datele necesare (valori de referinţă, limite de semnalizare, parametri de acordare ş.a.) fiind introduse în memoria calculatorului şi nu în elementele analogice individuale (care conform procedurii clasice sînt distribuite în fiecare circuit de reglare). În acest fel *DDC* oferă calculatoarelor de proces un cîmp larg de aplicabilitate pe seama reducerii numărului de aparate individuale.

Comanda şi reglarea numerică directă este un exemplu tipic de *conducere în timp real* cu divizarea timpului. Într-un astfel de sistem, variabilele

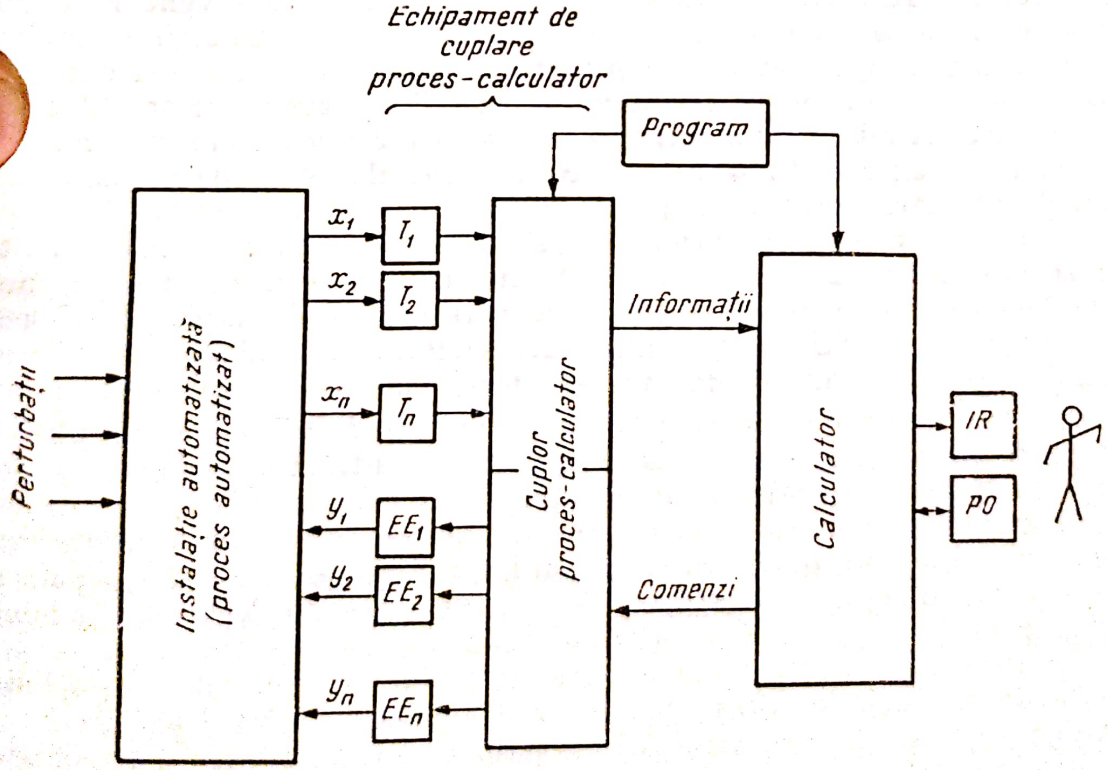


Fig. 10.44.

de intrare în calculator sînt mai întîi filtrate şi convertite succesiv în semnale numerice după un ciclu determinat de calculator. *Filtrarea* semnalelor se face cu ajutorul unor filtre cu constante de timp care variază între 0,5 secunde la procese relativ lente şi 0,01 s la procese rapide, sau apelînd la alte mijloace, cum ar fi conversia tensiune — frecvenţă, calculul mediei ş.a.

Conversia semnalelor analogice de intrare în semnale numerice se face odată cu eşantionarea semnalelor respective. În teoria informaţiei se demonstrează că la eşantionarea semnalelor analogice pierderea de informaţie este neglijabilă dacă frecvenţa de eşantionare este mai mare decît dublul frecvenţei (benzii) de trecere a spectrului frecvenţelor de măsurare. Aşadar, frecvenţa de eşantionare trebuie aleasă în deplină concordanţă cu dinamica procesului condus. Ea poate varia de la cîteva zeci de testări pe oră la procese relativ lente cum ar fi procesele calorice, de schimb de masă ş.a. la cîteva sute sau chiar mii de testări pe secundă la procese rapide cum ar fi cele electrice, ionice, electronice etc.

Cînd calculatorul se conectează la una din intrări, automat se conectează sincron şi la ieşirea sau ieşirile corespunzătoare intrării respective. Conectarea la ieşirile calculatorului se face deci în aceleaşi ritm în care se face testarea variabilelor de intrare, întreg sistemul acţionînd ca un regulator numeric multiplu cu performanţe şi supleţe înalte, conferite de programabilitatea uşoară a funcţiilor acestuia.

Comparînd cele două concepte, reies evident importante avantaje ale *DDC* şi anume:

— supleţea acestor sisteme în a îndeplini diverse funcţii;
— performanţe funcţionale superioare;
— comoditate în exploatare ş.a.

Într-adevăr, *DDC* oferă posibilitatea folosirii tehnicii moderne a calculatoarelor pentru a realiza orice lege de reglare, nu numai legile *P*, *I*, *PI*, *PD* şi *PID* ca la *SRA* convenţionale, precum şi orice strategie de reglare şi conducere, în speţă acea lege şi strategie care corespunde cel mai bine caracteristicilor statice şi dinamice ale instalaţiei automatizate. Aceasta se poate realiza relativ uşor printr-o programare corespunzătoare şi nu prin schimbarea echipamentului, ceea ce conferă calculatorului de proces caracterul de utilizare universală.

În ceea ce priveşte performanţele *DDC* este evident că orice calculator numeric are o precizie mai mare decît orice element similar analogic, fiind în acelaşi timp în măsură să furnizeze personalului de exploatare o mai mare cantitate de informaţie, prin diverse forme ca: „listinguri" (scrise de imprimante), semnale auditive, sub formă de grafice ş.a.

Un calculator de proces programat pentru *DDC* poate îndeplini şi funcţii de supraveghere a procesului şi echipamentului tehnologic, uşurînd astfel sarcinile personalului de conducere.

Eficienţa economică a utilizării unui sistem *DDC* este determinată de mai mulţi factori, dintre care cei mai importanţi sînt: costul echipamentului, costul întreţinerii acestuia şi beneficiile materiale legate de îmbunătăţirea performanţelor echipamentului tehnologic.

În timp ce costul echipamentului de reglare individuală creşte liniar cu numărul circuitelor de reglare, costul echipamentului *DDC* creşte cu atît mai puţin cu cît creşte numărul acestor circuite. Practica a demonstrat că sistemele *DDC* care preiau peste 100 bucle de reglare sau circuite de

Scanned with OKEN Scanner

comandă încep să devină mai economice, atît din punct de vedere al investiţiilor cît şi al exploatării, decît sistemele convenţionale de dată mai veche. Aceasta se explică prin înlocuirea a numeroase elemente individuale (regulatoare, indicatoare, convertoare, echipament de comandă numerică ş.a.) de către calculator, precum şi prin exploatarea mai comodă a acestui echipament unitar de înaltă calitate decît numeroasele aparate şi dispozitive cu funcţionare independentă.

În ceea ce priveşte introducerea *DDC* în diverse domenii, toţi utilizatorii recunosc că, după o perioadă de punere la punct, sistemele *DDC* asigură performanţe şi beneficii superioare sistemelor tradiţionale individuale. Aceasta se explică prin precizia, rapiditatea şi supleţea net superioare ale sistemelor *DDC*. Tocmai de aceea aceste sisteme capătă în ultima vreme o extindere din ce în ce mai mare.

Utilizarea sistemelor *DDC* ridică însă şi o serie de probleme legate de siguranţa în funcţionare. Concentrarea funcţiilor într-un singur element — calculatorul — şi utilizarea acestuia succesiv în diversele circuite informaţionale şi de comenzi „on line" prin divizarea timpului de lucru implică asigurarea unei totale siguranţe în exploatare, pentru că altfel scoaterea din funcţiune a calculatorului pune în pericol întreaga instalaţie condusă.

Realizările tehnologice din ultima vreme confirmă o îmbunătăţire remarcabilă a fiabilităţii calculatoarelor de proces, mai ales a elementului său de bază — unitatea centrală. S-au fabricat şi se exploatează calculatoare electronice miniaturizate cu circuite integrate şi echipament de interfaţă avînd o fiabilitate foarte mare, adică timpi medii de lucru între două defecţiuni de ordinul a zece-douăzeci de mii de ore.

Cînd ne referim la fiabilitatea unor astfel de echipamente este bine să acceptăm şi noţiunea de fiabilitate parţială. De exemplu, în multe cazuri defectarea unui circuit de reglare din sutele de astfel de circuite conectate la calculator nu înseamnă defectarea întregului sistem. Acesta poate funcţiona normal cu celelalte funcţiuni ale sale fără a afecta securitatea întregii instalaţii. Trebuie, deci, să se definească genul şi limitele defecţiunilor care afectează securitatea instalaţiei şi care nu permit continuarea desfăşurării normale a procesului tehnologic. De asemenea, trebuie să se definească şi genul şi limitele erorilor neadmise precum şi a celor admise pe scurte perioade tranzitorii.

În condiţiile în care echipamentul de conducere nu prezintă o siguranţă totală în exploatare se recurge la diverse alternative de îmbunătăţire a acesteia cum ar fi:

— înzestrarea instalaţiei de conducere cu dispozitive de comandă manuală sau cu dispozitive de reglare manuală care urmează a fi folosite în caz de defectare a sistemului de conducere;

— înzestrarea instalaţiei cu dispozitive de reglare automată independente de echipamentul de conducere cu calculator (dispozitive redondante), care urmează a fi folosite în caz de nevoie pentru buclele de reglare cele mai importante;

— utilizarea unui calculator de rezervă care în mod normal poate îndeplini alte funcţii (necondiţionate de timp, cum ar fi, de pildă, ţinerea evidenţei, optimizarea procesului ş.a.), iar în caz de defectare a calculatoru-

Scanned with OKEN Scanner

lui principal să preia funcţiile acestuia ca apoi, după remedierea defectului, să-şi schimbe rolurile, calculatorul de rezervă transmiţînd celui principal informaţia culeasă de la proces pe tot timpul funcţionării sale în aceste condiţii.

Optimizarea automată a activităţilor tehnice şi economice

Conducerea optimală sau *optimizarea* este stadiul cel mai avansat al conducerii prin care se asigură cele mai înalte performanţe ale procesului condus prin luarea în consideraţie a tuturor interacţiunilor şi implicaţiilor semnificative dintre variabilele procesului şi ale mediului înconjurător.

Optimizarea unui proces constă în elaborarea şi realizarea în practică a acelui algoritm de conducere care asigură valoarea extremă (maximă sau minimă) a funcţiei de performanţă cu precizia necesară şi volumul de operaţii cît mai mic. Aceasta comportă două aspecte esenţiale care în cele mai multe cazuri constituie şi două etape care se succed în mod continuu şi anume:

— căutarea prin calcul pe model sau prin explorare directă pe proces a direcţiei în care se află starea optimă faţă de starea curentă a procesului,

— organizarea şi executarea comenzilor necesare menite să conducă procesul din starea sa curentă către starea optimă.

Pentru aprecierea stării optime se utilizează un anumit criteriu: funcţia de performanţă. De aceea una din cerinţele cele mai importante ale conducerii optimale constă în alegerea şi specificarea corectă a acestei funcţii, căreia urmează să i se asigure o valoare extremă.

Noţiuni generale. Optimizarea se poate realiza prin două căi principal distincte:

— prin căutarea şi realizarea directă a optimului pe însăşi procesul în exploatare fără a utiliza în acest scop un model:

— prin simularea procesului pe un calculator pentru a determina performanţele optime posibile şi comenzile care asigură aceste performanţe, ca apoi acestea să fie executate asupra procesului.

În primul caz optimizarea se face experimental — pe proces, pe cînd în al doilea caz ea se face pe un model, urmînd ca ea să fie transpusă pe original.

Prima alternativă se recomandă pentru optimizarea proceselor insuficient cunoscute pentru a evita dificultăţile legate de elaborarea unui model. A doua alternativă se recomandă pentru optimizarea proceselor care pot fi modelate şi simulate cu suficientă precizie şi uşurinţă.

Optimizarea pe model prezintă avantajul că poate fi realizată mult mai rapid şi mai uşor decît pe proces fără ca procesul să fie perturbat artificial. Prezintă însă şi dezavantajul dificultăţilor legate atît de elaborarea modelului cît şi de actualizarea acestuia în cazul cînd caracteristicile procesului modelat se modifică în timp.

Evident, pot fi adoptate ambele căi simultan, recurgînd la optimizarea pe model a unor aspecte ce pot fi modelate şi la optimizarea pe original a altor aspecte greu sau imposibil de modelat.

Scanned with OKEN Scanner

În funcţie de dinamica procesului optimizat, adică în funcţie de viteza de răspuns a acestuia la diverse perturbaţii şi de frecvenţa şi intensitatea perturbaţiilor se recurge la:

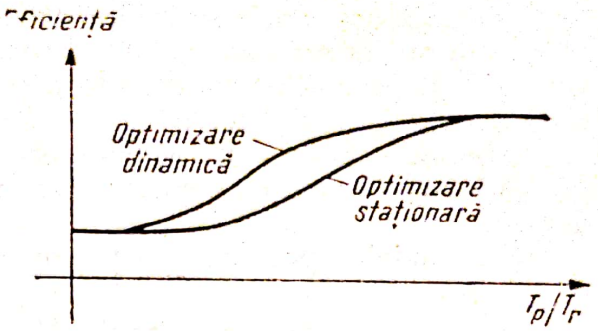


Fig. 10.45.

— optimizarea staţionară în cazul în care frecvenţa perturbaţiilor este relativ mică faţă de dinamica procesului, iar timpul nu apare ca variabilă a procesului;

— optimizarea dinamică, în caz contrar.

În figura 10.45 este prezentată o diagramă care arată eficienţa celor două moduri de optimizare în funcţie de raportul T_p/T_r, unde T_p este intervalul mediu între două perturbaţii, iar T_r timpul de răspuns al procesului la o perturbaţie de intensitate medie.

La valori mici ale acestui parametru (T_p/T_r), eficienţa ambelor moduri de optimizare este scăzută. Aceasta este situaţia proceselor cu constante de timp mari, care constituind filtre „trece jos", atenuează efectele frecvenţelor înalte şi manifestă inerţie la aceste frecvenţe.

La valori mari ale acestui parametru ambele moduri de optimizare au aceeaşi eficienţă ridicată deoarece perturbaţiile sînt relativ rare, iar răspunsul relativ rapid. La valori intermediare ale parametrului T_p/T_r optimizarea dinamică asigură o eficienţă superioară optimizării staţionare.

Atît optimizarea staţionară cît şi cea dinamică se pot realiza, după caz, fie în sisteme cu circuit închis, fie în sisteme cu circuit deschis, fie în sisteme combinate.

Toate procedeele de optimizare, dar mai ales optimizarea dinamică a proceselor complexe necesită un mare volum de calcule şi raţionamente care trebuie efectuate rapid şi frecvent, adică în timp util. O astfel de sarcină nu poate fi executată în „timp real" decît cu ajutorul calculatorului electronic. Numai acesta poate lua în consideraţie întregul ansamblu al variabilelor semnificative ale procesului supus optimizării, ca pe această bază să determine comenzile optime în conformitate cu criteriul de performanţă adoptat şi să le indice operatorului sau să le execute direct.

Metodele şi algoritmii de optimizare se diferenţiază de regulă, după modul de căutare a valorii extreme a funcţiei de performanţă care este legat sau chiar condiţionat de modul de acţiune al sistemului de optimizare, menit să conducă procesul din starea sa curentă în starea optimă. După acest criteriu metodele de optimizare pot fi clasificate după cum urmează:

— metode bazate pe programarea matematică;
— metode bazate pe calculul variaţional;
— metode adaptive bazate pe exploatarea evolutivă.

În cele ce urmează se vor avea în vedere metodele de determinare a unui optimum local cel mai apropiat de punctul de plecare. De aceea presupunem că funcţia de performanţă nu are decît un extremum sau că ne interesează optimul local.

Scanned with OKEN Scanner

Optimizarea staționară fără restricții. În acest subcapitol se prezintă problema optimizării staționare în sisteme deschise (cu acțiune după perturbație) a proceselor reprezentate prin modele neliniare (extremale) precise, nesupuse la nici un fel de restricții.

Abordăm mai întîi aspecte esențiale legate de căutarea optimului proceselor extremale cu o singură variabilă de stare și o variabilă de conducere.

Fie modelul procesului sub forma generală:

$$y = f(x, v), \tag{10—94}$$

unde y este variabila de stare, x — variabila de conducere, iar v — o variabilă perturbatoare.

Fie o funcție de performanță extremală de forma:

$$F = F(\bar{y}, \bar{x}, y, x, v), \tag{10—95}$$

unde $\bar{y}$ și $\bar{x}$ sînt valorile optime ale variabilelor y, respectiv x.

Prin înlocuirea lui y din (10—94) în (10—95) se obține funcția de performanță:

$$F_c = F_c(\bar{y}, \bar{x}, x, v). \tag{10—96}$$

Condiția existenței unui minimum al acestei funcții este

$$\frac{\partial F_c}{\partial x} = 0 \quad \text{și} \quad \frac{\partial^2 F_c}{\partial x^2} > 0, \tag{10—97}$$

iar condiția existenței unui maximum:

$$\frac{\partial F_c}{\partial x} = 0 \quad \text{și} \quad \frac{\partial^2 F_c}{\partial x^2} < 0. \tag{10—98}$$

Dacă și prima și a doua derivată sînt nule avem de-a face cu o nedeterminare și nu putem preciza dacă funcția are vreun extremum. Din condiția: $\partial F_c/\partial x = 0$ se determină expresia variabilei de conducere care asigură atingerea extremului funcției de performanță.

Ex. 10—13. Fie modelul procesului sub forma:

$$y = a_0 + a_1 x + bv \tag{10—99}$$

și fie funcția de performanță supusă minimizării:

$$F = C_y(\bar{y} - y)^2 + C_x(\bar{x} - x)^2, \tag{10—100}$$

unde C_y și C_x sînt constante.

Înlocuind pe y din (10—99) în (10—100) se obține:

$$F_c = C_y(\bar{y} - a_0 - a_1 x - bv)^2 + C_x(\bar{x} - x)^2. \tag{10—101}$$

Condiția de minimum a lui F_c este:

$$\frac{\partial F_c}{\partial x} = -2C_y a_1(\bar{y} - a_0 - a_1 x - b v) - 2C_x(\bar{x} - x) = 0 \tag{10—102}$$

$$\frac{\partial F_c^2}{\partial x^2} = 2C_y a_1^2 + 2C_x > 0. \tag{10—103}$$

Scanned with OKEN Scanner

Scanned with OKEN Scanner

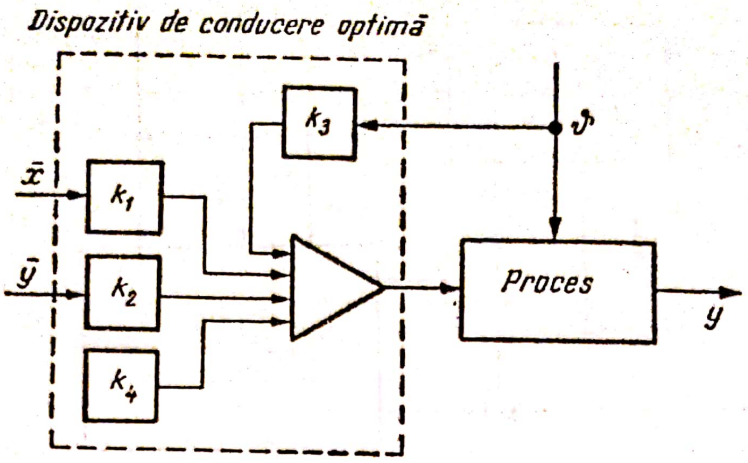


Fig. 10.46.

Explicitînd (10—102) în x se obţine ecuaţia conducerii optimale:

$$x = \frac{C_x \bar{x} + C_y a_1 \bar{y} - C_y a_1 b v - C_y a_0 a_1}{C_y a_1^2 + C_x} \qquad (10-104)$$

sau

$$x = k_1 \bar{x} + k_2 \bar{y} + k_3 v + k_4, \qquad (10-104')$$

unde:

$$k_1 = \frac{C_x}{C_y a_1^2 + C_x}; \quad k_2 = \frac{C_y a_1}{C_y a_1^2 + C_x}; \quad k_3 = \frac{-C_y a_1 b}{C_y a_1^2 + C_x}; \quad k_4 = \frac{-C_y a_0 a_1}{C_y a_1^2 + C_x}.$$

Schema conducerii optimale a unui astfel de proces în sistem cu circuit deschis este prezentată în figura 10.46.

Ex.10—14. Fie: $C_y = C_x = 1$; $a_0 = 1$; $a_1 = 2$;

$\bar{y} = 5$; $\bar{x} = 3$;

$b = 2$; $v = 1$

În acest caz $k_1 = 0{,}2$; $k_2 = 0{,}2$; $k_3 = -0{,}8$; $k_4 = -0{,}4$. Prin urmare:

$x = 0{,}2 \cdot 3 + 0{,}4 \cdot 5 - 0{,}8 \cdot 1 - 0{,}4 = 0{,}4 = 1{,}4.$

Posibilităţile de optimizare staţionară a unui proces extremal cu două variabile de intrare, x şi v, sînt ilustrate în figura 10.47. Diversele valori ale variabilelor $x-v$ pentru care funcţia y are aceeaşi valoare sînt unite în planul $x-v$ prin contururi închise.

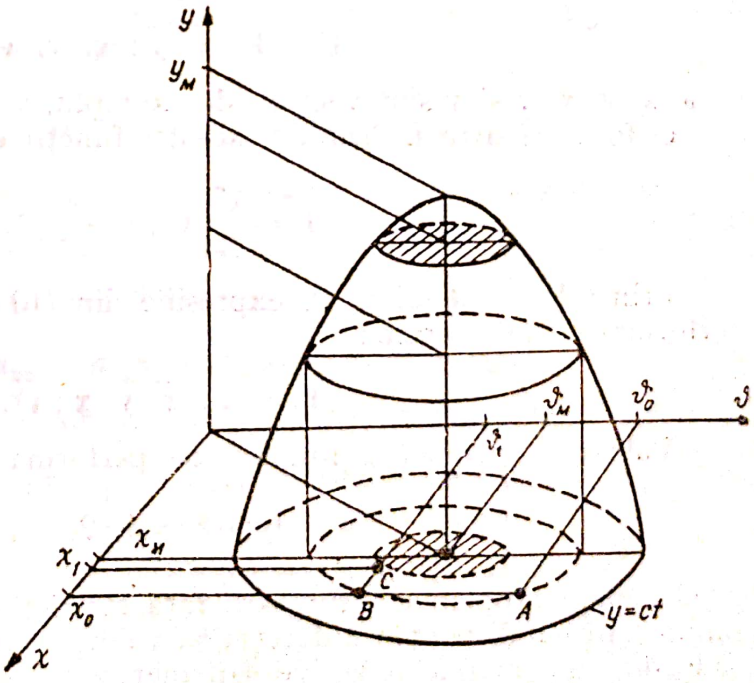


Fig. 10.47.

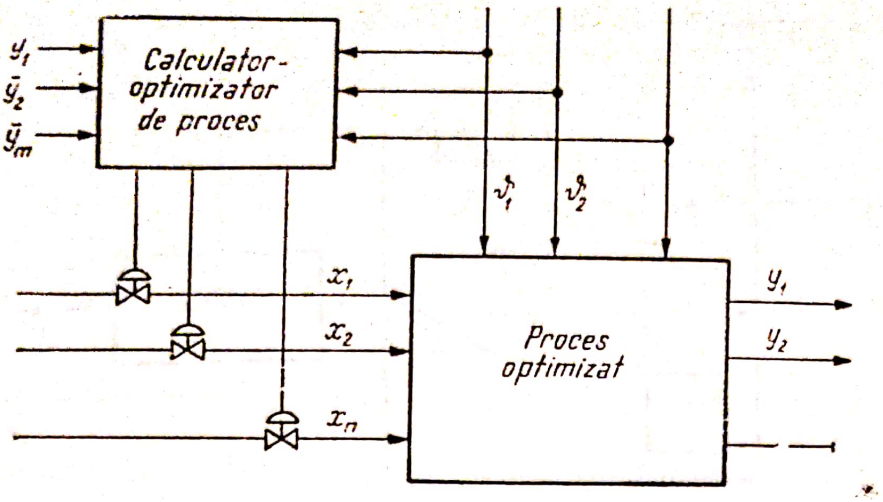


Fig. 10.48.

Să presupunem că în momentul inițial $x = x_0$, iar $v = v_0$, adică o stare corespunzătoare punctului A. Dacă x nu-și modifică valoarea, iar v variază de la v_0 la v_1, y are la început o creștere apoi o scădere — punctul B. Dacă în continuare v rămîne constant, iar x variază de la x_0 la x_1, y va căpăta o valoare sporită, corespunzătoare punctului c. Valoarea maxim maximorum a lui y poate fi atinsă dacă variabilele x și v capătă valorile x_M, respectiv v_M, corespunzătoare punctului M.

Să luăm acum în considerație un proces cu n variabile independente de comandă, r variabile perturbatoare și m variabilele de stare (fig. 10.48)

Modelul procesului este exprimat de sistemul de ecuații

$$\mathbf{y}_i = \mathbf{y}_i(x_1, x_2 \ldots, x_n, v_1, v_2, \ldots, v_r) \quad i = 1,2, \ldots m \qquad (10-105)$$

iar funcția de performanță poate fi pusă sub forma generală;

$$\mathbf{F} = \mathbf{F}\ (\bar{\mathbf{x}}, \bar{\mathbf{y}}; \mathbf{x}, \mathbf{y}, \mathbf{v}), \qquad (10-106)$$

unde $\mathbf{x}$, $\mathbf{y}$, $\mathbf{x}$, $\mathbf{y}$ și $\mathbf{v}$ sînt vectori, de exemplu, $\mathbf{x} = \mathbf{x}(x_1, x_2, \ldots, x_n)$.

O formă foarte utilizată a acestei funcții este forma patratică:

$$\mathbf{F} = \sum_{i=1}^{m} \mathrm{C}_i(\bar{y}_i - y_i)^2. \qquad (10-107)$$

Prin înlocuirea lui y_i cu expresiile din (10—105) se obține funcția de performanță sub forma:

$$\mathbf{F}_c = \mathbf{F}_c\ (\bar{\mathbf{x}}, \bar{\mathbf{y}}, \mathbf{x}; \mathbf{v}). \qquad (10-108)$$

Valoarea extremă a funcției de performanță se obține din condiția

$$\frac{\partial F_c}{\partial x_k} = 0; \quad k = 1, 2, \ldots, n; \qquad (10-109)$$

prin urmare problema optimizării fără restricții constă în rezolvarea sistemului (10—109) pentru a determina valorile optime ale lui x_k folosind metode directe, metode iterative sau metode experimentale pe model — simularea.

Scanned with OKEN Scanner

În cazul modelelor liniare de forma

$$\begin{aligned}
y_1 &= a_{10} + a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n + b_{11}v_1 + b_{12}v_2 + \ldots + b_{1r}v_r \\
y_2 &= a_{20} + a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n + b_{21}v_1 + b_{22}v_2 + \ldots + b_{2r}v_r \\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
y_m &= a_{m0} + a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n + b_{m1}v_1 + b_{m2}v_2 + \ldots + b_{mr}v_r
\end{aligned} \tag{10—110}$$

realizarea conducerii optimale depinde de numărul variabilelor de stare, adică de m şi de numărul variabilelor de comandă, adică de n.

Sistemul care are un număr de variabile de stare egal cu numărul variabilelor de comandă, adică $n = m$ se numeşte sistem determinat. Dacă $n > m$ sistemul este supradeterminat, iar dacă $n < m$ sistemul este subdeterminat. Analizăm în continuare aceste trei cazuri.

Cazul sistemelor determinate, $n = m$. Fie un proces de forma:

$$\begin{aligned}
y_1 &= a_{10} + a_{11}x_1 + a_{12}x_2 + b_{11}v_1 + b_{12}v_2 \\
y_2 &= a_{20} + a_{21}x_1 + a_{22}x_2 + b_{21}v_1 + b_{22}v_2
\end{aligned} \tag{10—111}$$

unde $n = m = r = 2$.

Fie funcţia de performanţă sub forma:

$$\mathbf{F} = \sum_{i=1}^{2} C_i(\bar{y}_i - y_i)^2 \tag{10—112}$$

Înlocuind (10—111) în (10—112) se obţine:

$$\begin{aligned}
\mathbf{F}_c = {} & C_1(\bar{y}_1 - a_{10} - a_{11}x_1 - a_{12}x_2 - b_{11}v_1 - b_{12}v_2)^2 + \\
& C_2(\bar{y}_2 - a_{20} - a_{21}x_1 - a_{22}x_2 - b_{21}v_1 - b_{22}v_2)^2.
\end{aligned} \tag{10—113}$$

Derivînd în raport cu x_1 şi respectiv cu x_2, se obţine un sistem de două ecuaţii — ecuaţiile conducerii optimale:

$$\begin{aligned}
\frac{\partial F}{\partial x_1} = {} & -2C_1a_{11}(\bar{y}_1 - a_{10} - a_{11}x_1 - a_{12}x_2 - b_{11}v_1 - b_{12}v_2) + \\
& -2C_2a_{21}(\bar{y}_2 - a_{20} - a_{21}x_1 - a_{22}x_2 - b_{21}v_1 - b_{22}v_2) = 0,
\end{aligned} \tag{10—114}$$

$$\begin{aligned}
\frac{\partial F}{\partial x_2} = {} & -2C_2a_{12}(\bar{y}_1 - a_{10} - a_{11}x_1 - a_{12}x_2 - b_{11}v_1 - b_{12}v_2) + \\
& -2C_2a_{22}(\bar{y}_2 - a_{20} - a_{21}x_1 - a_{22}x_2 - b_{21}v_1 - b_{22}v_2) = 0.
\end{aligned}$$

Acest sistem poate fi pus sub următoarea formă matriceală:

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \bar{y}_1 - a_{10} - b_{11}v_1 - b_{12}v_2 \\ \bar{y}_2 - a_{20} - b_{21}v_1 - b_{22}v_2 \end{pmatrix}, \tag{10—115}$$

de unde se poate calcula valorile celor două variabile de conducere.

Cazul sistemelor supradeterminate, $n > m$. În acest scop $(n - m)$ variabile de conducere pot primi valori arbitrare, de exemplu — valori nule, iar m variabile de conducere urmează să primească valori determinate din ecuaţiile conducerii optimale.

Scanned with OKEN Scanner

Fie $m = r = 1$ şi $n = 2$, iar modelul procesului — de forma:

$$y_1 = a_{10} + a_{11}x_1 + a_{12}x_2 - b_{11}v_1. \tag{10—116}$$

Fie funcţia de performanţă

$$F = C_1(\bar{y}_1 - y_1)^2 = C_1(y - a_{10} - a_{11}x_1 - a_{12}x_2 - b_{11}v_1)^2. \tag{10—117}$$

Derivînd (10—117) în raport cu x_1 şi respectiv cu x_2, se obţine:

$$\begin{aligned} \frac{\partial F}{\partial x_1} &= -2C_1a_{11}(\bar{y}_1 - a_{10} - a_{11}x_1 - a_{12}x_2 - b_{11}v_1) = 0, \\ \frac{\partial F}{\partial x_2} &= -2C_1a_{12}(\bar{y}_1 - a_{10} - a_{11}x_1 - a_{12}x_2 - b_{11}v_1) = 0. \end{aligned} \tag{10—118}$$

Se observă că cele două ecuaţii sînt dependente. Prin urmare dispunem de o singură ecuaţie care poate fi explicitată în x_1 sau în x_2:

$$x_1 = (\bar{y}_1 - a_{10} - a_{12}x_2 - b_{11}v_1)/a_{11}. \tag{10—119}$$

Cazul sistemelor subdeterminate, $n < m$. În acest caz nu se poate realiza decît o conducere suboptimală deoarece valoarea funcţiei de performanţă nu poate fi redusă pînă la zero.

Fie $m = 3$, $r = 1$ şi $n = 2$, iar modelul procesului — de forma:

$$\begin{aligned} y_1 &= a_{10} + a_{11}x_1 + a_{12}x_2 - b_{11}v \\ y_2 &= a_{20} + a_{21}x_1 + a_{22}x_2 + b_{21}v \\ y_3 &= a_{20} + a_{31}x_1 + a_{32}x_2 + b_{31}v \end{aligned} \tag{10—120}$$

Fie funcţia de performanţă

$$F = \sum_{i=1}^{3} C_i(\bar{y}_i - y_i). \tag{10—121}$$

Înlocuind (10—120) în (10—121), se obţine:

$$\begin{aligned} F_c = {} & C_1(\bar{y}_1 - a_{10} - a_{11}x_1 - a_{12}x_2 - b_{11}v)^2 + \\ & C_2(\bar{y}_2 - a_{20} - a_{21}x_1 - a_{22}x_2 - b_{21}v)^2 + \\ & C_3(\bar{y}_3 - a_{30} - a_{31}x_1 - a_{32}x_2 - b_{32}v)^2. \end{aligned} \tag{10—122}$$

Derivînd (10—122) în raport cu x_1 şi respectiv cu x_2, se obţine:

$$\begin{aligned} \frac{\partial F_c}{\partial x_1} = {} & -2C_1a_{11}(\bar{y}_1 - a_{10} - a_{11}x_1 - a_{12}x_2 - b_{11}v) + \\ & -2C_2a_{21}(\bar{y}_2 - a_{20} - a_{21}x_1 - a_{22}x_2 - b_{21}v) + \\ & -2C_3a_{31}(\bar{y}_3 - a_{30} - a_{31}x_1 - a_{32}x_2 - b_{32}v) = 0 \\ \frac{\partial F_c}{\partial x_2} = {} & -2C_1a_{12}(\bar{y}_1 - a_{10} - a_{11}x_1 - a_{12}x_2 - b_{11}v) + \\ & -2C_2a_{22}(\bar{y}_2 - a_{20} - a_{21}x_1 - a_{22}x_2 - b_{21}v) + \\ & -2C_3a_{32}(\bar{y}_3 - a_{30} - a_{31}x_1 - a_{32}x_2 - b_{31}v) = 0. \end{aligned} \tag{10—123}$$

Comparînd (10—114) cu (10—123) se constată că nu poate fi vorba de acelaşi rezultat datorită apariţiei celui de al treilea termen al sumei în

Scanned with OKEN Scanner

(10—123). De altfel se poate demonstra că un proces extremal cu m variabile de stare nu poate fi condus optimal dacă nu dispunem de un număr de variabile de comandă cel puţin egal cu m.

În general este greu şi uneori imposibil să se realizeze o conducere optimală a proceselor tehnologice moderne. În afară de faptul că acestea sînt multivariabile, în practică intervin erorile de măsurare şi calcul ca şi unele aspecte dinamice ale procesului. Inerţia, întîrzierile şi erorile cu care se obţin valorile variabilelor procesului şi cu care se execută comenzile necesare optimizării au ca efect o conducere suboptimală. Un calculator conectat la proces şi programat corespunzător poate îmbunătăţi substanţial situaţia prin calcule de optimizare frecvente şi precise.

Optimizarea staţionară cu restricţii. Majoritatea proceselor supuse optimizării sînt procese cu restricţii. La unele din acestea valoarea optim optimorum nu poate fi atinsă din cauza unor restricţii de exploatare a instalaţiilor tehnologice (limite de temperatură, presiune, debit etc.), de achiziţii a materialelor necesare, de desfacere a produselor fabricate ş. a. La acestea se pune problema realizării valorii extreme a funcţiei de performanţă pentru restricţiile (condiţiile) date.

Astfel de probleme se rezolvă cu metode specifice cum ar fi : programarea liniară şi neliniară, programarea dinamică ş. a.

Problema optimizării staţionare cu restricţii constă, în principiu, în determinarea valorii extreme, maximă sau minimă, a funcţiei de performanţă, adică :

$$\min/\max F = \min/\max F(x_1, x_2, \ldots, x_n ; v_1, v_2, \ldots, v_r) \quad (10—124)$$

unde : $x_1, x_2, \ldots, x_n$ sînt variabile comandate ;
$v_1, v_2, \ldots, v_r$ — variabile necomandate ;
$F(x,v)$ este funcţie de performanţă ;

respectînd restricţiile privind domeniile de variaţie admisibile pentru variabilele procesului şi/sau pentru diverse combinaţii ale acestora, adică respectînd inecuaţiile :

$$R_k(x_1, x_2, \ldots, x_n) > L_k ; \quad (k = 1, 2, \ldots, p), \qquad (10—125)$$

unde $L_1\ L_2, \ldots L_p$ sînt limite admisibile ;
precum şi restricţiile privind realizabilitatea fizică a procesului, adică ;

$$x_k > 0 ; \quad (k = 1, 2, \ldots, n). \qquad (10—126)$$

Rezolvarea acestui gen de probleme cu metoda programării liniare şi neliniare este prezentată în lucrarea noastră [12] la capitolul 6, paginile 11—105 şi respectiv în capitolul 7, pag. 105—145, unde sînt prezentate şi numeroase exemple de calcul.

Metoda multiplicatorilor Lagrange.

În acest subcapitol avem în vedere optimizarea proceselor neliniare (funcţii de performanţă neliniare) cu modele precise şi restricţii (liniare sau neliniare) sub forma unor relaţii între variabilele funcţiei de performanţă.

Scanned with OKEN Scanner

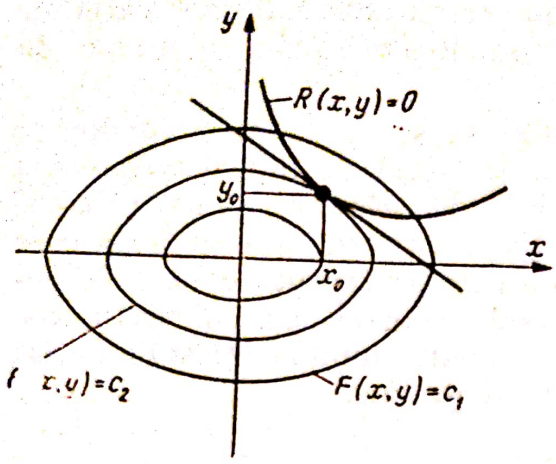


Fig. 10.49.

Pentru înțelegerea metodei să ne referim la figura 10—49 unde se prezintă graficul cu diversele valori ale funcției de performanță $F = F(x, y) = ct.$ și graficul restrictei neliniare $R(x, y) = 0$.

Fie M punctul optim pentru care $R(x,y) = R(x_0,y_0) = 0$ și $F(x,y) = F(x_0,y_0)$. Se observă că în punctul optim tangenta la curba $F(x,y)$ coincide cu tangenta la curba $R(x,y)$. Ecuația acestei tangente se poate exprima astfel:

$$(x - x_0)\left.\frac{\partial F(x, y)}{\partial x}\right|_0 + (y - y_0)\left.\frac{\partial F(x, y)}{\partial y}\right|_0 = 0; \qquad (10-127)$$

$$(x - x_0)\left.\frac{\partial R(x, y)}{\partial x}\right|_0 + (y - y_0)\left.\frac{\partial F(x, y)}{\partial y}\right|_0 = 0. \qquad (10-128)$$

Împărțind (10—127) la (10—128) și regrupînd termenii se obține:

$$\frac{\left.\frac{\partial F(x, y)}{\partial x}\right|_0}{\left.\frac{\partial R(x, y)}{\partial x}\right|_0} = -\frac{\left.\frac{\partial F(x, y)}{\partial y}\right|_0}{\left.\frac{\partial R(x, y)}{\partial y}\right|_0} = \lambda. \qquad (10-129)$$

Raportul (10—129) luat cu semn schimbat se numește multiplicator Langrange. Folosind această noțiune ecuațiile (10—127) și (10—128) pot fi transcrise sub forma:

$$\left.\frac{\partial F(x, y)}{\partial x}\right|_0 + \lambda\left.\frac{\partial R(x, y)}{\partial x}\right|_0 = 0; \qquad (10-130)$$

$$\left.\frac{\partial F(x, y)}{\partial y}\right|_0 + \lambda\left.\frac{\partial R(x, y)}{\partial y}\right|_0 = 0. \qquad (10-131)$$

Ecuațiile (10—130), (10—131) constituie condițiile necesare ale optimului funcției F în prezenta restricției R. Aceste ecuații împreună cu ecuația de restricție servesc pentru determinarea conducerii optimale: x_0, y_0 și λ.

Metoda poate lua în considerație mai multe restricții. Astfel, în cazul general, al unei funcții de performanță de forma:

$$F_c(x_1, x_2, \ldots x_n) = F_c(x) \qquad (10-132)$$

și a p restricții de forma:

$$R_k(x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_m) = F(x, y) = 0; \quad k = 1, 2, \ldots, p \qquad (10-133)$$

putem construi funcția Lagrange

$$F_L(x) = F_c(x) + \sum_{k=1}^{p} \lambda_k R_k(x, y). \qquad (10-134)$$

Scanned with OKEN Scanner

Derivînd (10—134) în raport cu x, se obţine un sistem de n ecuaţii de forma:

$$\frac{\partial F_L}{\partial x_i} = \frac{\partial F_c}{\partial x_i} + \sum_{k=1}^{p} \lambda_k \frac{\partial R_k}{\partial x_i} = 0\,;\ i = 1, 2, \ldots, n, \quad (10-135)$$

care urmează să fie rezolvat împreună cu cele p ecuaţii ale restricţiilor (10—133), formînd un sistem de $n + p$ ecuaţii cu $n + p$ necunoscute:

$$x_1, x_2, \ldots, x_n, \lambda_1, \lambda_2, \ldots, \lambda_p.$$

În rezolvarea acestui sistem adesea se întîmpină serioase dificultăţi de calcul. Pentru depăşirea acestora se poate proceda în felul următor:

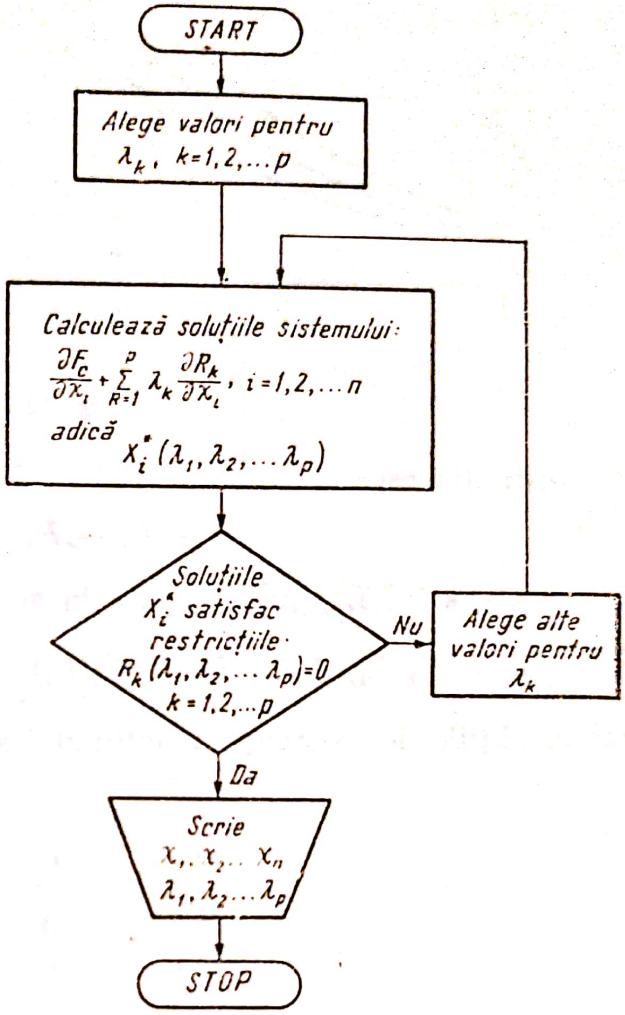


Fig. 10.50.

Se aleg anumite valori pentru λ_k $(k = 1, 2, \ldots p)$ şi cu aceste valori se rezolvă sistemul de n ecuaţii cu n necunoscute x_i $(i = 1, 2, \ldots n)$. Soluţiile astfel obţinute care depind de valorile alese pentru λ sînt apoi folosite în ecuaţiile celor p restricţii. Dacă aceste valori satisfac şi cele p restricţii problema este rezolvată, în caz contrar se aleg alte valori pînă cînd valorile lui λ satisfac atît ecuaţiile (10—134) cît şi ecuaţiile (10—135).

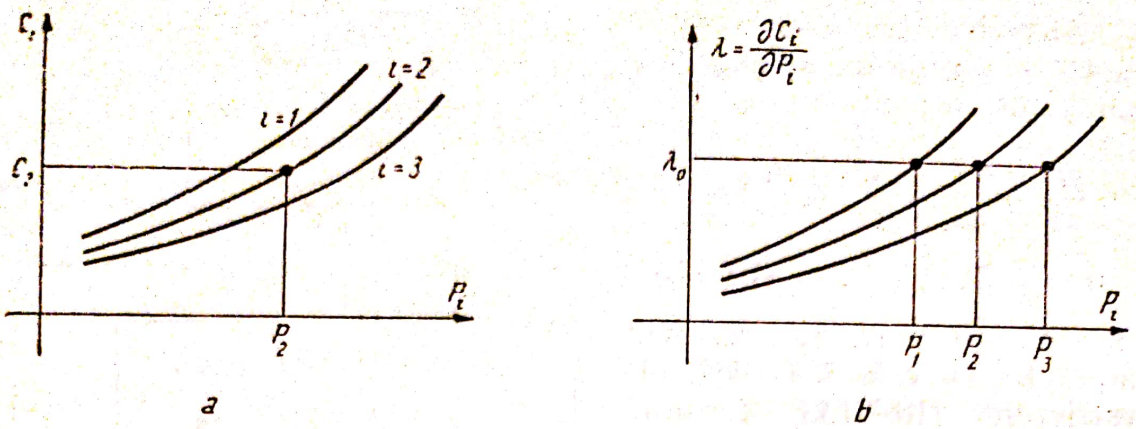
Procedura de rezolvare a acestor probleme este prezentată în figura 10.50, de unde se vede că rezolvarea sistemului neliniar de $n + p$ ecuaţii cu $n + p$ necunoscute este transformată în rezolvarea unui sistem de p ecuaţii cu p necunoscute, $\lambda_1, \lambda_2, \ldots, \lambda_p$, unde la fiecare iteraţie în precizarea valorilor lui λ se rezolvă sistemul de ecuaţii (10—135). Această procedură poate fi executată de către un calculator numeric.

Ex. 10—15. Fie o baterie de trei reactoare chimice care funcţionează în paralel. Cheltuielile specifice $C_i (i = 1,2,3)$ pentru unitatea de produs realizat de fiecare reactor cresc cu productivitatea P_i a reactorului după o curbă ca cea din figura 10.51, *a*, iar raportul dintre variaţia cheltuielilor de producţie şi variaţia productivităţii fiecărui reactor, adică $\partial C_i/\partial P_i$, variază după curbele din figura 10.51, *b*.

Se cere să se determine cheltuielile minime pentru exploatarea celor trei reactoare care urmează să realizeze o producţie totală dată P şi productivităţile optime ale fiecărui reactor. Prin urmare funcţia de performanţă care trebuie minimizată este

$$F = C_1(P_1) + C_2(P_2) + C_3(P_3), \quad (10-136)$$

Scanned with OKEN Scanner



a *b*

Fig. 10.51.

iar restricţia este

$$P - P_1 - P_2 - P_3 = 0. \qquad (10-137)$$

Ecuaţia lui Lagrange capătă în acest caz forma:

$$F_L = C_1(P_1) + C_2(P_2) + C_3(P_3) + \lambda(P - P_1 - P_2 - P_3) \qquad (10-138)$$

iar condiţiile de existenţă a optimului sînt:

$$\frac{\partial F_L}{\partial P_1} = \frac{\partial C_1(P_1)}{\partial P_1} - \lambda = 0$$

$$\frac{\partial F_L}{\partial P_2} = \frac{\partial C_2(P_2)}{\partial P_2} - \lambda = 0 \qquad (10-139)$$

$$\frac{\partial F_L}{\partial P_3} = \frac{\partial C_3(P_3)}{\partial P_3} - \lambda = 0.$$

Din condiţiile de optimum rezultă că cele trei reactoare trebuie să funcţioneze la aceeaşi valoare a raportului $\partial C/\partial P$, adică:

$$\frac{\partial C_1(P_1)}{\partial P_1} = \frac{\partial C_2(P_2)}{\partial P_2} = \frac{\partial C_3(P_3)}{\partial P_3} = \lambda. \qquad (10-140)$$

Cheltuielile minime de exploatare a celor trei reactoare se determină prin iteraţii dînd diverse valori variabile λ, astfel încît să satisfacă restricţia (10—137) aşa cum reiese şi din organigrama din figura 10.52.

Optimizarea staţionar-dinamică evolutivă

Metodele evolutive se folosesc îndeosebi pentru optimizarea proceselor pentru care nu se poate stabili cu exactitate un model adecvat. În astfel de cazuri se recurge la determinarea directă a optimului pe însăşi procesul optimizat prin metode interative cu ajutorul cărora variabilele de conducere se modifică (corectează) pe baza analizei efectului produs asupra performanţelor procesului de modificările anterioare.

Scanned with OKEN Scanner

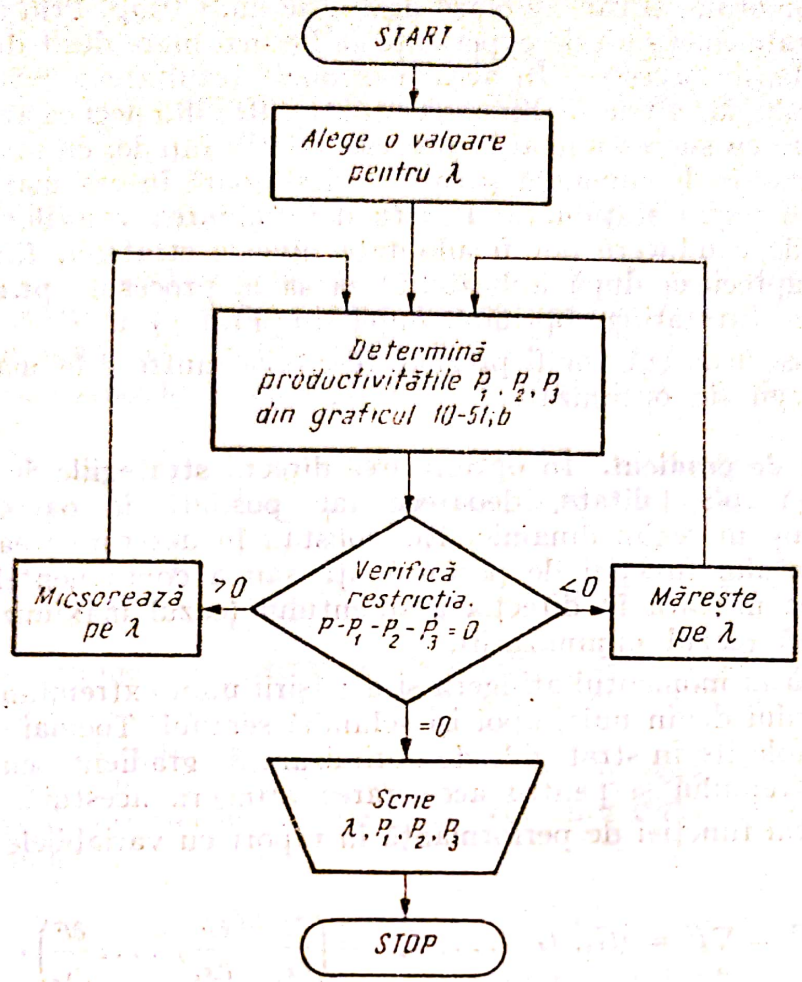


Fig. 10.52.

Aceste metode pot fi aplicate și asupra modelelor. În acest caz simularea se execută, de obicei, la o scară de timp diferită de 1, pe cînd dacă se aplică asupra procesului, experimentul de optimizare se realizează în timp real, adică la scara de timp 1.

Optimizarea evolutivă directă poate fi într-o anumită măsură și adaptivă. Ea poate fi realizată de către un calculator de proces care, pe baza unui program sau subprogram de optimizare modifică în mod organizat valorile variabilelor de conducere ale procesului (sau ale modelului) și analizează efectul acestor modificări, adică variația valorii funcției de performanță.

Corelînd efectul cu cauza care l-a produs determină apoi care modificări au ameliorat valoarea funcției de performanță și care au înrăutățit-o. Pe baza acestei analize calculatorul poate determina, organiza și efectua noi comenzi menite să amelioreze situația în cel mai scurt timp posibil, alternînd comenzile cu analiza efectului și noi comenzi.

Fără a contesta avantajele optimizării directe pe proces, care ocolește dificultățile legate de elaborarea unui model, această optimizare are și părțile sale slabe. Pentru optimizarea unui proces cu multe variabile comandate este necesar să se organizeze un mare număr de experiențe care se fac în

Scanned with OKEN Scanner

prezența zgomotului și care durează destul de mult timp. Prin urmare, s-ar putea ca durata unei serii de experiențe să fie mai mare decît durata dintre două perturbații succesive. În acest caz unele rezultate vor fi cauzate de situația inițială, iar altele — din noua situație. Rezultă deci că aceste metode pot fi aplicate cu succes numai la procese relativ rapide, cu un număr moderat de variabile de comandă și care se desfășoară în cea mai mare parte a timpului în regim staționar. Pentru determinarea valorilor optime ale variabilelor de conducere pot fi adoptate diverse strategii. Eficiența unei strategii se apreciază după aplicabilitatea sa la procesul optimizat, după rapiditatea cu care atinge optimul, după simplitate ș. a.

În cele ce urmează vor fi prezentate cîteva dintre cele mai uzuale și simple strategii de optimizare.

Strategii de gradient. În optimizarea directă strategiile de gradient au cea mai largă aplicabilitate, deoarece fac posibilă în oarecare măsură și o optimizare în regim dinamic. Ele constau în determinarea pe diverse căi a gradientului funcției de performanță sau a componentelor acestuia și organizarea mișcării în direcția gradientului (cazul maximizării) sau în direcția opusă (cazul minimizării).

Se știe că în momentul atingerii și depășirii unui extremum componentele gradientului devin nule, apoi își schimbă semnul. Tocmai aceste proprietăți sînt folosite în strategiile de optimizare de gradient pentru a sesiza atingerea extremului și pentru accelerarea atingerii acestuia.

Gradientul funcției de performanță în raport cu variabilele comandate este

$$G = \nabla F = (G_1, G_2, \ldots, G_n) = \left(\frac{\partial F}{\partial x_1}, \frac{\partial F}{\partial x_2}, \ldots, \frac{\partial F}{\partial x_n}\right), \qquad (10-141)$$

unde $G_1 = \frac{\partial F}{\partial x_1} \ldots, G_n = \frac{\partial F}{\partial x_n}$ sînt componentele acestuia.

Modulul gradientului este

$$M = (G_1^2 + G_2^2 + \ldots, G_n^2)^{1/2}, \qquad (10-142)$$

iar vectorul director este

$$g = \frac{G}{M} = \frac{1}{M}(G_1, G_2, \ldots, G_n). \qquad (10-143)$$

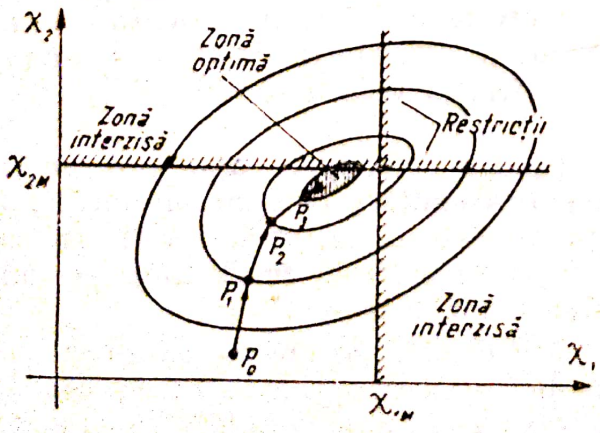


Fig. 10.53.

Înaintarea către optimum (extremum) se face, de regulă, în pași pe o anumită traiectorie. O traiectorie bună trebuie să aducă procesul din orice stare inițială într-o stare optimă într-un timp cît mai scurt, nedepășind zona limitată de restricții.

În figura 10.53 este prezentată o traiectorie în planul $x_1 - x_2$ cu res-

Scanned with OKEN Scanner

tricţii realizată în trei paşi, P_0P_1, P_1P_2 şi P_2P_3, punctul P_3 fiind considerat final deoarece se află în zona optimă.

Fiecare pas al traiectoriei este caracterizat printr-o direcţie şi printr-un modul. Relaţia de recurenţă pentru vectorul variabilelor de conducere la pasul următor lui p este:

$$x_{p+1} = x_p + \Delta x_p = x_p + H_p g_p, \qquad (10-144)$$

unde x_p şi g_p sînt vectori coloană pentru pasul p, H_p este o matrice diagonală a coeficienţilor modulului pentru acelaşi pas, iar $H_p g_p$ este corecţia variabilelor de conducere pentru pasul următor $(p+1)$.

Pentru n variabile de conducere relaţia (10—144) capătă forma:

$$\begin{pmatrix} x_{1(p+1)} \\ x_{2(p+1)} \\ \cdots\cdots \\ x_{n(p+1)} \end{pmatrix} = \begin{pmatrix} x_{1p} \\ x_{2p} \\ \cdots \\ x_{np} \end{pmatrix} + \begin{pmatrix} H_{1p}\,0 \ldots 0 \\ 0\,H_{2p} \ldots 0 \\ \cdots\cdots\cdots \\ 0\;0 \quad \ldots \quad H_{np} \end{pmatrix} \cdot \begin{pmatrix} g_{1p} \\ g_{2p} \\ \cdots \\ g_{np} \end{pmatrix} \qquad (10-145)$$

Se cunosc numeroasele strategii de gradient care se deosebesc prin modul de calcul al corecţiei variabilelor de conducere. Aici vom prezenta ca exemplu numai două.

Strategia Cauchy. Conform acestei strategii corecţia variabilelor de conducere se determină în paşi pe baza vectorilor directori g, după următoarea formulă:

$$\Delta x_p = x_{p+1} - x_p = H_p g_p, \qquad (10-145)$$

unde H_p este o matrice diagonală cu coeficienţii constanţi pentru toţi paşii. Organigrama acestei strategii este dată în figura 10.54.

Această strategie se caracterizează printr-o viteză relativ redusă de atingere a optimului, deoarece la fiecare pas se determină atît valoarea funcţiei de performanţă, cît şi vectorul director al gradientului. În apropierea extremului determinarea acestui vector ca şi constatarea atingerii acestuia devin dificile, mai ales în prezenţa zgomotului.

Mărimea pasului, adică valoarea coeficienţilor matricei H, trebuie să fie suficient de mare pentru a realiza o viteză acceptabilă de apropiere de maximum, dar nu prea mare pentru a nu depăşi punctul sau zona optimă.

Rezultă deci că o astfel de strategie poate fi utilizată în faza iniţială a optimizării, urmînd ca în faza finală a atingerii optimului să fie adoptată o altă strategie.

Strategia gradientului iniţial. Aceasta se deosebeşte de precedenta prin faptul că la fiecare pas se determină numai valoarea funcţiei de performanţă, determinarea gradientului director urmînd să se facă numai în momentul cînd funcţia de performanţă atinge sau depăşeşte o valoare extremă.

Astfel, dacă plecînd dintr-o anumită stare în direcţia gradientului iniţial către un maximum, se constată o creştere a funcţiei de performanţă,

tricţii realizată în trei paşi, P_0P_1, P_1P_2 şi P_2P_3, punctul P_3 fiind considerat final deoarece se află în zona optimă.

Fiecare pas al traiectoriei este caracterizat printr-o direcţie şi printr-un modul. Relaţia de recurenţă pentru vectorul variabilelor de conducere la pasul următor lui p este:

$$x_{p+1} = x_p + \Delta x_p = x_p + H_p g_p, \qquad (10-144)$$

unde x_p şi g_p sînt vectori coloană pentru pasul p, H_p este o matrice diagonală a coeficienţilor modulului pentru acelaşi pas, iar $H_p g_p$ este corecţia variabilelor de conducere pentru pasul următor $(p+1)$.

Pentru n variabile de conducere relaţia (10—144) capătă forma:

$$\begin{pmatrix} x_{1(p+1)} \\ x_{2(p+1)} \\ \cdots\cdots \\ x_{n(p+1)} \end{pmatrix} = \begin{pmatrix} x_{1p} \\ x_{2p} \\ \cdots \\ x_{np} \end{pmatrix} + \begin{pmatrix} H_{1p}\,0 \ldots 0 \\ 0\,H_{2p} \ldots 0 \\ \cdots\cdots\cdots \\ 0\;0 \quad \ldots \; H_{np} \end{pmatrix} \cdot \begin{pmatrix} g_{1p} \\ g_{2p} \\ \cdots \\ g_{np} \end{pmatrix} \qquad (10-145)$$

Se cunosc numeroasele strategii de gradient care se deosebesc prin modul de calcul al corecţiei variabilelor de conducere. Aici vom prezenta ca exemplu numai două.

Strategia Cauchy. Conform acestei strategii corecţia variabilelor de conducere se determină în paşi pe baza vectorilor directori g, după următoarea formulă:

$$\Delta x_p = x_{p+1} - x_p = H_p g_p, \qquad (10-145)$$

unde H_p este o matrice diagonală cu coeficienţii constanţi pentru toţi paşii. Organigrama acestei strategii este dată în figura 10.54.

Această strategie se caracterizează printr-o viteză relativ redusă de atingere a optimului, deoarece la fiecare pas se determină atît valoarea funcţiei de performanţă, cît şi vectorul director al gradientului. În apropierea extremului determinarea acestui vector ca şi constatarea atingerii acestuia devin dificile, mai ales în prezenţa zgomotului.

Mărimea pasului, adică valoarea coeficienţilor matricei H, trebuie să fie suficient de mare pentru a realiza o viteză acceptabilă de apropiere de maximum, dar nu prea mare pentru a nu depăşi punctul sau zona optimă.

Rezultă deci că o astfel de strategie poate fi utilizată în faza iniţială a optimizării, urmînd ca în faza finală a atingerii optimului să fie adoptată o altă strategie.

Strategia gradientului iniţial. Aceasta se deosebeşte de precedenta prin faptul că la fiecare pas se determină numai valoarea funcţiei de performanţă, determinarea gradientului director urmînd să se facă numai în momentul cînd funcţia de performanţă atinge sau depăşeşte o valoare extremă.

Astfel, dacă plecînd dintr-o anumită stare în direcţia gradientului iniţial către un maximum, se constată o creştere a funcţiei de performanţă,

Scanned with OKEN Scanner

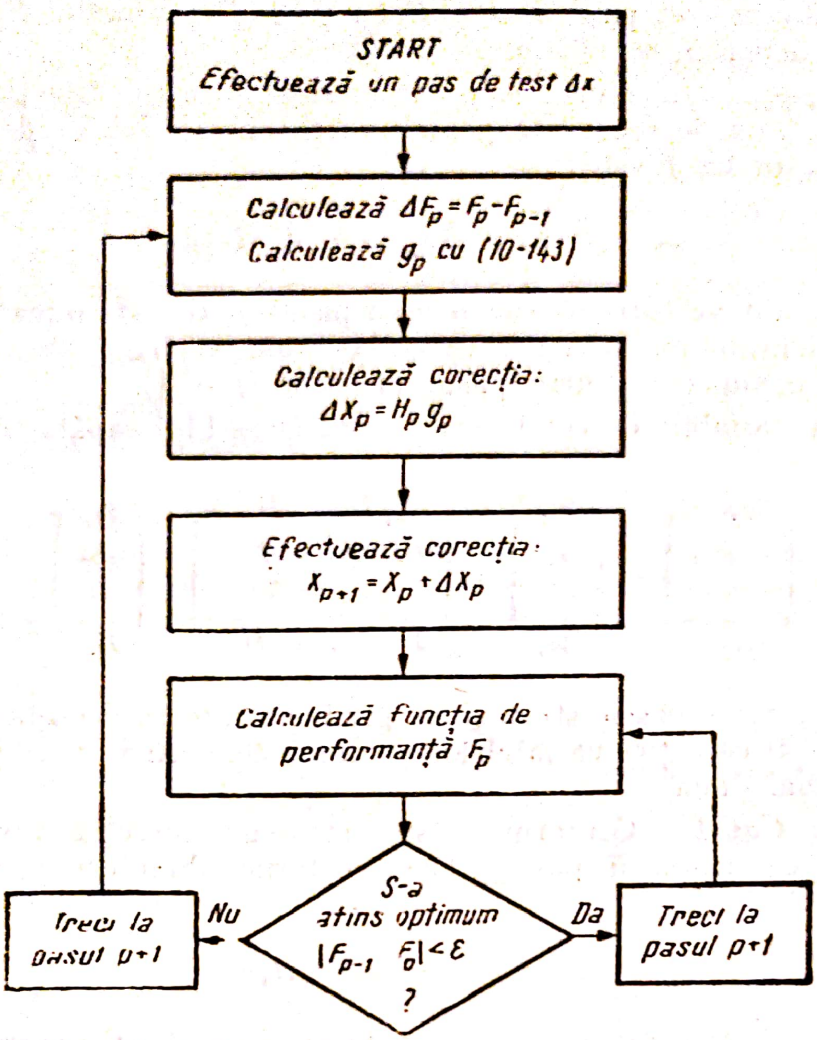


Fig. 10.54.

la pașii următori se continuă deplasarea în direcția gradientului inițial pînă cînd se constată o micșorare a funcției de performanță față de valoarea precedentă. Din acel moment se suprimă sau se reduce la jumătate înaintarea în direcția gradientului inițial și se organizează înaintarea în direcția unui nou gradient ș. a. m. d. pînă se atinge punctul sau zona optimă.

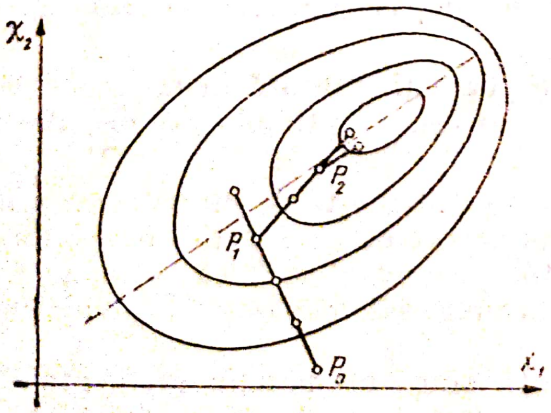


Fig. 10.55.

În figura 10.55 este ilustrat un exemplu de optimizare într-un sistem cu două variabile de conducere. După al treilea pas în direcția gradientului inițial din punctul P_0 se determină un nou gradient ș. a. m. d.

Prin această strategie se ajunge mai repede în starea optimă decît prin strategia precedentă, dar și aceasta este deficitară în ce privește atingerea ei, deoarece o poate depăși și

oscila în jurul ei. O ameliorare în acest sens s-ar obține dacă în apropierea zonei optime durata sau viteza pașilor s-ar micșora.

Determinarea practică a gradientului. Dacă dispunem de modelul procesului optimizat sub forma unor ecuații, gradientul se obține pe baza derivatelor parțiale ale funcției de performanță în raport cu variabilele de conducere.

Dacă optimizarea se face direct pe proces, gradientul se poate determina pe baza răspunsului procesului la mici pași de testare folosind relații cu diferențe finite pentru aproximarea derivatelor parțiale în jurul punctelor de evaluare.

Cea mai simplă relație cu diferențe, care aproximează derivata parțială a funcției F în raport cu variabila de conducere x_i, este

$$\frac{\partial F}{\partial x_i} \simeq \left(\frac{\Delta F}{\Delta x_i}\right)_p ; \quad i = 1, 2 \ldots, n, \qquad (10-146)$$

unde $\Delta F = F(x_1, x_2 \ldots x_i + \Delta x_i, \ldots, x_n) - F(x_1, x_2, \ldots x_i, \ldots x_n)$, iar p marchează punctul de evaluare.

Un mod mai corect de a aproxima această mărime constă în considerarea variației lui x de ambele părți ale punctului p, adică adoptînd un punct de evaluare central față de care se consideră variația $\pm \Delta x_i/2$. În acest caz F se determină din relația:

$$F = F(x_1, x_2, \ldots x_i + \Delta x_i/2, \ldots x_n) - F(x_1, x_2, \ldots x_i - \Delta x_i/2, \ldots x_n). \qquad (10-147)$$

Aceste relații se remarcă prin simplitate dar sînt deficitare prin faptul că nu reprezintă media mai multor determinări ceea ce face ca zgomotul să influențeze negativ asupra preciziei mai ales în zona gradienților mici. În literatura de specialitate sînt descrise o serie de metode perfecționate care înlătură deficiența semnalată.

Optimizarea adaptivă

În unele cazuri parametrii unor procese variază în timp, iar legile de desfășurare a acestora depind de diverse condiții de lucru. Conducerea optimă a acestor procese presupune actualizarea destul de frecventă a modelului (inclusiv a funcției de performanță) la condițiile de lucru create și posibilitatea adaptării unor strategii și tactici de optimizare diferite în funcție de condițiile de lucru date. O astfel de conducere se realizează în sisteme adaptive care ar putea fi caracterizate astfel:

Sistemele adaptive sînt sisteme automate care au aptitudinea de a-și autoajusta structura și/sau funcțiunile în concordanță cu schimbarea parametrilor procesului optimizat și schimbarea condițiilor externe de lucru, astfel încît să îndeplinească un anumit criteriu de optimalitate.

Optimizarea adaptivă se realizează în sisteme în care valorile variabilelor unui proces în desfășurare sînt relevate de către un *identificator* care utilizează aceste valori pentru reactualizarea modelului sau pentru reevaluarea stării procesului, ca pe această bază să se ia o *decizie* și să se efectueze *comenzile* necesare în vederea atingerii stării optime pentru condițiile

Scanned with OKEN Scanner

date. Prin urmare, optimizarea adaptivă are trei funcţii esenţiale: identificarea, decizia şi comanda.

Funcţia de identificare constă în obţinerea şi prelucrarea centralizată a informaţiei referitoare la proces astfel încît să servească la luarea unei decizii şi/sau la reactualizarea modelului.

Funcţia de decizie constă în evaluarea stării procesului direct sau pe model după anumite reguli şi criterii în vederea alegerii unui plan de acţiune adecvat situaţiei curente.

Funcţia de comandă constă în realizarea în practică a planului de acţiune adoptat cum ar fi, de exemplu, modificarea valorilor de referinţă ale unor regulatoare, modificarea parametrilor de acordare a acestora sau modificarea directă a anumitor variabile de conducere.

Toate aceste funcţiuni pot fi îndeplinite total sau parţial de către un calculator de proces.

Optimizarea adaptivă se realizează, de regulă, în sisteme combinate (cu acţiune după abatere şi după perturbaţie) cu mai multe circuite. Circuitul cel mai interior serveşte pentru identificarea şi reactualizarea modelului, iar cel sau cele exterioare servesc pentru optimizare. În figura 10—56 este prezentată schema generală simplificată a unui sistem optimal adaptiv.

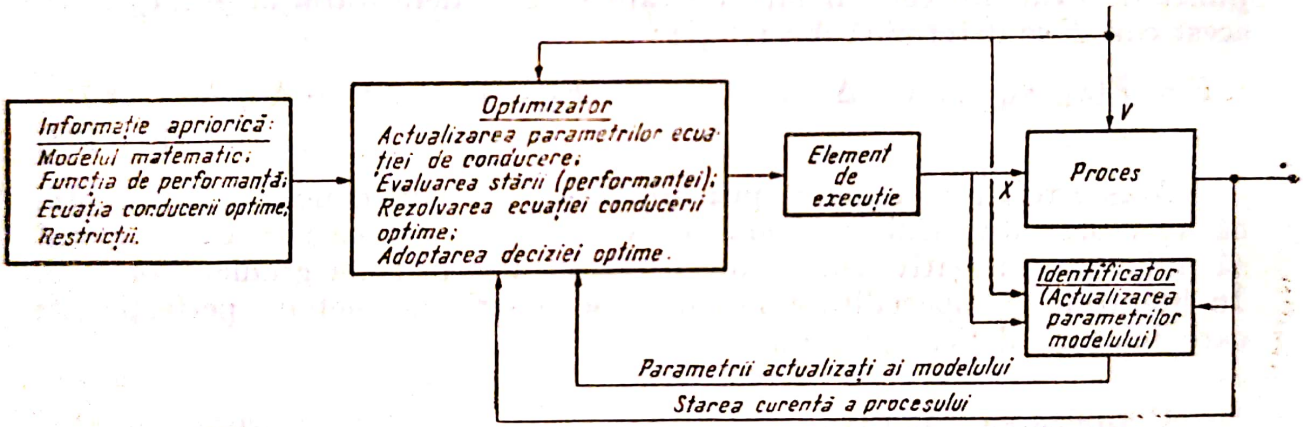


Fig. 10.56.

Adaptarea la schimbarea parametrilor procesului. Schimbarea parametrilor (coeficienţilor) unui model are loc pe măsură ce procesul modelat iese din limitele de liniaritate, fiind specifică deci proceselor neliniare. Adaptarea la situaţia existentă se poate face atît în sisteme deschise cît şi în sisteme închise (cu reacţie).

Adaptarea în acest caz constă în esenţă în actualizarea modelului, celelalte operaţii legate de evaluare funcţiei de performanţă şi de comandă a procesului efectuîndu-se aşa cum s-a arătat mai înainte.

Să considerăm un model liniar cu m variabile de stare şi n variabile de conducere de forma:

$$y_i = \sum_{j=1}^{n} a_{ji}\, x_i + \sum_{k=1}^{r} b_{ki}\, v_k; \qquad i = 1, 2, \ldots, m, \qquad (10-148)$$

Scanned with OKEN Scanner

care este supus conducerii optime după funcţia de performanţă:

$$F = F(\bar{y}_1 \bar{y}_2, \ldots, \bar{y}_m; y_1, y_2, \ldots, y_m, v_1, v_2, \ldots, v_r, x_1, x_2, \ldots, x_n). \quad (10-149)$$

Cu valorile variabilelor y, x şi v corespunzătoare pasului p şi cu cele corespunzătoare a q paşi precedenţi se alcătuieşte o tabelă — (tabela 10—2) — care ar putea fi memorată şi actualizată de către un calculator de proces. În acest scop la fiecare nou pas se renunţă la datele cele mai vechi, adică la cele corespunzătoare pasului $p-q$, se deplasează datele anterioare de pe o linie mai veche spre una mai nouă, adică spre începutul tabelei, iar ultima linie — linia corespunzătoare pasului curent p — se completează cu datele cele mai actuale.

Tabela 10—2

Nr. crt.	Variabila x	Variabila v	Variabila y	Funcţia de performanţă F
	$x_1 \ldots x_m$	$v_1 \ldots v_r$	$y_1 \ldots y_m$	
$p-q$	$x_1(p-q) \ldots x_m(p-q)$	$v_1(p-q) \ldots v_r(p-q)$	$y_1(p-q) \ldots y_m(p-q)$	$F(p-q)$
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
$p-2$	$x_1(p-2) \ldots x_m(p-2)$	$v_1(p-2) \ldots v_r(p-2)$	$y_1(p-2) \ldots y_m(p-2)$	$F(p-2)$
$p-1$	$x_1(p-1) \ldots x_m(p-1)$	$v_1(p-1) \ldots v_r(p-1)$	$y_1(p-1) \ldots y_m(p-1)$	$F(p-1)$
p	$x_1(p) \ldots x_m(p)$	$v_1(p) \ldots v_r(p)$	$y_1(p) \ldots y_m(p)$	$F(p)$

Pentru aprecierea măsurii în care modelul corespunde procesului modelat se poate recurge la compararea funcţiei de performanţă F_p, cores-

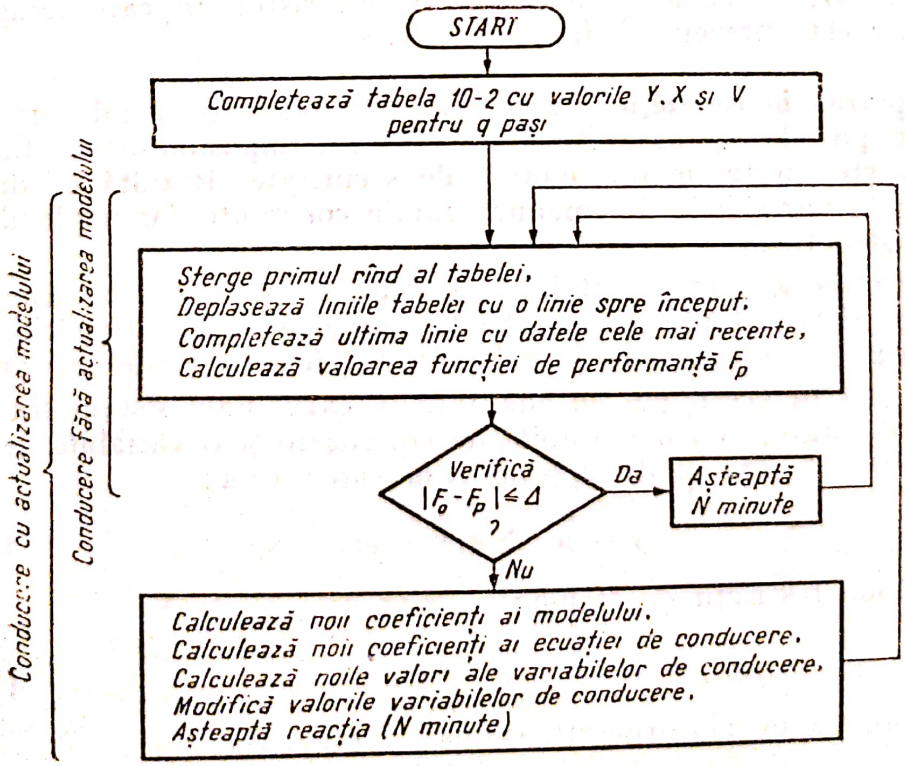


Fig. 10.57.

Scanned with OKEN Scanner

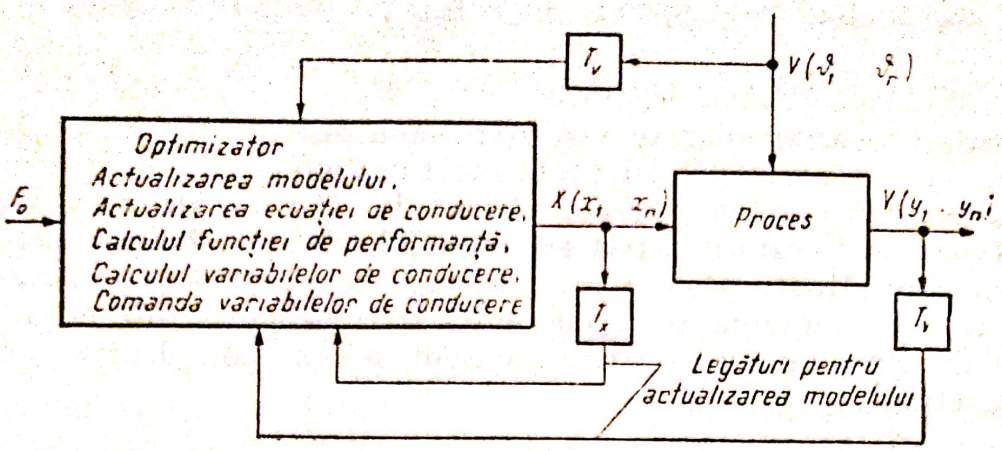


Fig. 10.58.

punzătoare pasului p, cu o valoare optimă F_0, dinainte prescrisă. Modelul se consideră corespunzător, iar procesul — bine condus dacă la fiecare nou pas

$$|F_0 - F_p| \leqslant \Delta, \tag{10—150}$$

unde Δ este o eroare limită admisă. În caz contrar atît modelul cît și conducerea sînt considerate necorespunzătoare, iar pentru a le ameliora urmează să se mărească precizia modelului printr-o reactualizare.

Reactualizarea modelului constă în reevaluarea coeficienților a_{ji} și b_{ki}, care se poate face prin metoda analizei prin regresie pe baza datelor din tabela 10.2. Organigrama unei astfel de proceduri este prezentată în figura 10.57, iar schema simplificată a unui sistem în care aceasta poate fi realizată este prezentată în figura 10.58.

Adaptarea la restricții. În general variabilele de conducere ale unui proces nu pot depăși anumite limite fie din imposibilitatea fizică de a depăși aceste limite fie din motive de securitate. Rezultă că dincolo de aceste limite variabilele de comandă rămîn constante, iar modelul are altă expresie (structură).

Restricțiile asupra variabilelor de stare și asupra variabilelor de conducere pot fi transformate în restricții numai asupra variabilelor de conducere prin introducerea în relațiile de restricție a modelului procesului.

Vom ilustra acest gen de adaptare în cazul unui sistem simplu cu o variabilă de stare, cu o variabilă de conducere și o variabilă necondusă.

Ex. 10—16. Fie modelul procesului sub forma:

$$y = a_0 + a_1 x + bv. \tag{10—151}$$

Fie două restricții de forma:

$$0 < x; \quad x \leqslant x_s. \tag{10—152}$$

Fie funcția de performanță supusă minimizării:

$$F = (\bar{y} - y)^2. \tag{10—153}$$

Scanned with OKEN Scanner

Înlocuind (10.151) în (10.153) se obține:

$$F = (\bar{y} - a_0 - a_1 x - bv)^2. \qquad (10-154)$$

Derivînd în raport cu x și egalînd cu zero se obține ecuația conducerii optime:

$$x = (\bar{y} - a_0 - bv)/a_1. \qquad (10-155)$$

Dar această variabilă este supusă restricțiilor (10—152). Înlocuind (10.155) în (10.152) se obține:

$$a_0 + bv \leqslant \bar{y} \leqslant a_0 + a_1 x_s + bv. \qquad (10-156)$$

Pentru determinarea soluției optime valoarea lui y este comparată cu limitele relației (10.156):

— dacă $\bar{y} < a_0 + bv$, soluția optimă este $x = 0$;
— dacă $a_0 + bv \leqslant \bar{y} \leqslant a_0 + a_1 x_s + bv$ soluția optimă este (10.155);
— dacă $\bar{y} > a_0 + a_1 x_s + bv$ soluția optimă este $x = x_s$.

S-ar putea spune că sistemul are trei modele de conducere pentru cele trei cazuri posibile.

Optimizarea dinamică

În paragrafele precedente s-au prezentat diverse metode de optimizare staționară în care timpul nu era luat în considerație. În continuare vom indica modul în care se abordează și soluționează o problemă de optim dinamic sau cum se mai spune — de optim general.

În regim staționar starea optimă poate fi determinată din condiția de extremum a unei funcții de performanță de forma:

$$F = F(\mathbf{y}, \mathbf{x}, \mathbf{v}), \qquad (10-157)$$

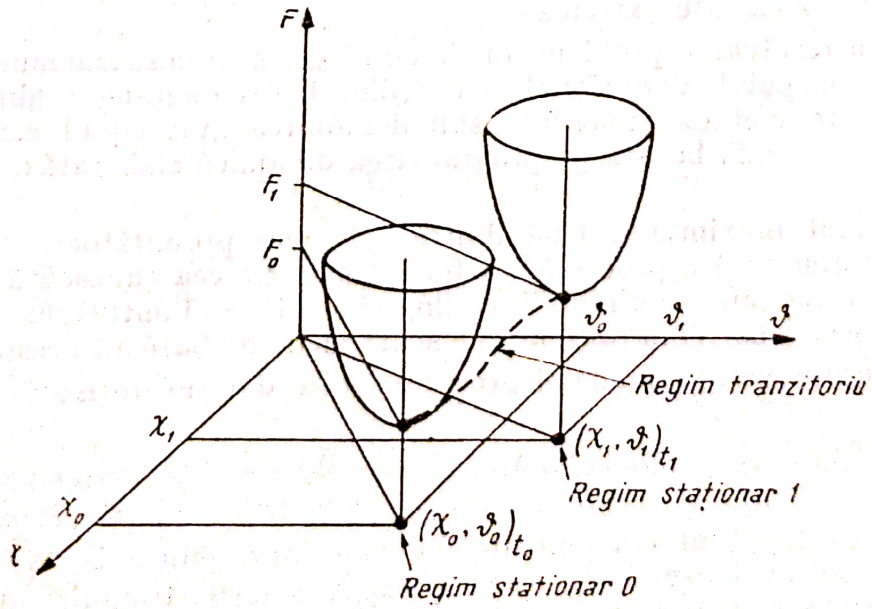


Fig. 10.59.

Scanned with OKEN Scanner

unde y este vectorul coloană m-dimensional al variabilelor de stare, x-vectorul coloană n-dimensional al variabilelor de conducere, iar v — vectorul r-dimensional al variabilelor necomandate.

Modificarea vectorului variabilelor necomandate $V(v_1, v_2, \ldots, v_r)$ duce la modificarea vectorului de stare $Y(y_1, y_2, \ldots, y_m)$ şi deci şi la modificarea stării (optime) — figura 10.59. În sistemele fizice această modificare are loc în timp. Pentru a conduce un sistem dintr-o stare 0 într-o nouă stare (optimă) 1 este necesar să se modifice în mod corespunzător vectorul variabilelor de conducere $X(x_1, x_2, \ldots, x_n)$. Dar, datorită inerţiei modificarea stării sistemelor nu se face instantaneu, ci în timp, prin urmare trebuie să se ia în consideraţie variabila independentă — timpul. Din această cauză modelul matematic al sistemelor dinamice trebuie să pună în evidenţă derivatele în raport cu timpul variabilelor de stare. Un astfel de model ar putea fi de forma:

$$\frac{dy_i}{dt} = f[y(t), x(t), v(t)], \quad i = 1, 2, \ldots m \qquad (10-158)$$

unde se păstrează notaţiile de mai înainte.

Pentru evaluarea eficienţei conducerii pe toată durata regimurilor tranzitorii dintre două stări staţionare (traiectoria punctată din figura 10.59) urmează să se folosească nu o funcţie, ci o funcţională, de forma:

$$\Phi = \int_{t_0}^{t_f} F[y(t), x(t), v(t)]\, dt, \qquad (10-159)$$

unde t_0 şi t_f este momentul de timp iniţial şi respectiv final.

Problema optimizării generale constă în determinarea în funcţie de condiţiile iniţiale a unui vector $x[x_1(t), x_2(t), \ldots, x_n(t)]$ care să conducă sistemul din starea iniţială într-o stare finală astfel încît funcţionala (10—159) să aibă o valoare extremă.

Pentru rezolvarea problemelor de acest gen se cunosc mai multe metode bazate pe calculul variaţional (ecuaţiile Euler-Lagrange) dintre care vom prezenta metoda cunoscută sub denumirea „principiul maximului", precum şi o metodă bazată pe programarea dinamică elaborată de Bellman.

Principiul maximului. Una dintre cele mai promiţătoare tehnici de optimizare dinamică a proceselor deterministe este cea cunoscută sub denumirea de „principiul maximului" elaborată de L. S. Pontreaghin şi perfecţionată de şcoala sa. Vom prezenta pe scurt ideile de bază ale acestei metode.

Să presupunem că modelul procesului este dat sub forma:

$$\frac{dy_i}{dt} = f_i(y_1, y_2, \ldots, y_m;\ x_1, x_2, \ldots, x_n;\ t);\ i = 1, 2, \ldots, m, \qquad (10-160)$$

unde $y_1, y_2, \ldots y_m$ sînt componentele vectorului de stare Y, $x_1, x_2, \ldots, x_n$ -componentele vectorului de conducere X, t este timpul, iar f_i sînt funcţii derivabile în raport cu x.

Scanned with OKEN Scanner

O anumită comandă (vector de conducere)

$$x(t) = x[x_1(t).\ x_2(t), \ldots, x_n(t)] \qquad (10-161)$$

poate „deplasa" sistemul (10—160) din starea inițială y_0 într-o anumită stare finală y_f.

Din mulțimea comenzilor realizabile urmează să se aleagă numai comenzi admise, adică variabilele de conducere urmează să primească valori cuprinse în interiorul și pe conturul unei zone denumite zona de conducere, definită de sistemul de restricții:

$$R_{oj} \leqslant x_i \leqslant R_{sj};\quad i = 1,2, \ldots, n \qquad (10-162)$$

și care poate fi reprezentată printr-un paralelipiped într-un spațiu n-dimensional.

Pentru fiecare comandă (vector de conducere) $x(t)$ sistemul evoluează după o anumită traiectorie. Aceasta se poate determina prin integrarea sistemului (10—160), adică prin găsirea unei soluții:

$$y(t) = y[y_1(t), y_2(t), \ldots, y_m(t)] \qquad (10-163)$$

cu respectarea restricțiilor (10.162).

În astfel de cazuri se pune problema ca într-un anumit interval de timp $(t_f - t_0)$ sistemul să fie condus din starea inițială $y_0[y_{10}, y_{20}, \ldots, y_{n0}]$, în care se află în momentul de timp t_0, în stare finală dorită $y_f[y_1, y_2, \ldots, \ldots, y_n]$ la momentul de timp t_f.

Pentru a evalua eficiența conducerii, fiecărei traiectorii admise, i se asociază o funcțională de forma:

$$\Phi = \int_{t_0}^{t_f} (Y, X)\, dt. \qquad (10-164)$$

În final se pune problema ca din mulțimea vectorilor de comandă $x(t)$ care în intervalul de timp $t_0 < t < t_f$ aduce procesul din starea inițială y_0 în starea finală y_f să se aleagă acela care asigură funcționalei (10 — 164) cea mai mică valoare posibilă.

Modelul (10—160) și funcționala (10—164) pot fi puse sub o formă comună:

$$\frac{dy_i}{dt} = F_i(Y, X);\quad i = 1, 2, \ldots, m. \qquad (10-165)$$

Sistemului (10—165) îi putem asocia un alt sistem de ecuații diferențiale în care intervin variabilele ajutătoare $\psi_0(t), \psi_1(t), \ldots \psi_n(t)$, adică sistemul:

$$\frac{d\psi_i}{dt} = \sum_{k=0}^{m} \frac{\partial F_k(Y, X)}{\partial y_i} \psi_k;\quad i = 1, 2, m. \qquad (10-166)$$

Ecuațiile (10—165) și (10—166) pot fi exprimate împreună cu ajutorul hamiltonianului $\mathcal{H}$ de variabile $x_0, x_1, \ldots, x_n, y_0, y_1, \ldots, y_m$ și $\psi_0, \psi_1, \ldots, \ldots, \psi_n$, adică

$$\mathcal{H}(X, Y, \psi) = \sum_{i=1}^{m} \psi_i F_i(X, Y). \qquad (10-167)$$

Scanned with OKEN Scanner

De remarcat că ecuațiile (10—165) și (10—166) pot fi puse sub forma:

$$\frac{dy_i}{dt} = \frac{\partial \mathcal{H}}{\partial \psi_i}; \quad i = 1, 2, \ldots, m \qquad (10—168)$$

$$\frac{d\psi_i}{dt} = -\frac{\partial \mathcal{H}}{\partial y_i}; \quad i = 1, 2, \ldots, m \qquad (10—169)$$

Prezentăm enunțul principiului maximului, fără a face demonstrația, care este foarte laborioasă:

O comandă $X(t)$ cu $t_0 < t < t_f$ este optimă, adică funcționala Φ va avea o valoare minimă, dacă se îndeplinesc următoarele condiții:

— pentru orice moment de timp $t_0 < t < t_f$ hamiltonianul $\mathcal{H}(X, Y, \psi)$ are o valoare maximă;

— pentru $t = t_f$ variabila auxiliară $\psi_0(t)$ este $\psi_0(t) \leqslant 0$.

Din nefericire dificultățile de calcul pentru rezolvarea acestor probleme cu metoda Pontreaghin ca și cu alte metode variaționale sînt foarte mari, ceea ce explică, într-o oarecare măsură, progresele relative modeste în aplicarea lor la procese industriale.

Programarea dinamică. Această metodă constă în folosirea unor relații de recurență bazate pe principiul optimalității, elaborat de Bellman, care permite ca problema optimizării pe un anumit interval de timp să fie transformată într-o problemă de optimizare succesivă pe intervale de timp mai mici.

Conform principiului optimalității traiectoria optimă are proprietatea că indiferent de condițiile și deciziile inițiale celelalte soluții sau decizii trebuie să fie soluții și decizii optime față de starea în care a ajuns procesul.

Aplicat la un model sub formă discretă cu N pași principiul optimalității poate fi formulat astfel:

Porțiunea din traiectoria optimă începînd cu pasul p și pînă la pasul N, corespunzătoare aducerii procesului din starea intermediară y_p în starea finală $y_f = y_N$, constituie traiectoria optimă a acestor pași.

Fie modelul procesului sub forma discretă

$$y_{p+1} = \psi_1[y_p, x_p, \Delta t] \qquad (10—157)$$

și funcționala de performanță

$$\Phi_{1\to N}^{opt} = \sum_{p=1}^{N} F[y_p, x_p], \qquad (10—158)$$

unde y_p este variabila de stare, iar x_p variabila de conducere în momentul de timp $t_0 + p\Delta t$.

Dacă din (10.158) se scoate primul termen de sub semnul sumei se obține:

$$\Phi_{1\to N}^{opt}(y_0) = \underset{x_0}{opt}\left[F(y_0, x_0) + opt \sum_{p=2}^{N} F_p(y_p, x_p)\right]. \qquad (10—159)$$

Să observăm că:

$$opt \sum_{p=2}^{N} F_p(x_p, x_p) = \Phi_{2\to N}^{opt}(y_1). \qquad (10—160)$$

Scanned with OKEN Scanner

Cu această notație relația de recurență devine:

$$\Phi^{opt}_{1\to N}(y_0) = {}^{opt}_{x_0}[F(y_0, x_0) + \Phi^{opt}_{2\to N}(y_1)]. \qquad (10-161)$$

Utilizînd modelul procesului care leagă variabila de stare de variabila de conducere se poate stabili o nouă formă a relației de recurență în care să intervină numai variabila x, adică:

$$\Phi^{opt}_{1\to N}(y_0) = {}^{opt}_{x_0}[F^{opt}_c(x_0) + \Phi^{opt}_{2\to N}(y_1)]. \qquad (10-162)$$

Pe această bază se deduce relația de recurență directă pentru pasul p, sub forma:

$$\Phi^{opt}_{1\to N}(y_0) = F^{opt}_{c_1}(x_0) + F^{opt}_{c_2}(x_1) + \ldots + F_{c(p-1)}(x_{p-2}) + \Phi^{opt}_{p\to N}(x_{p-1}), \qquad (10-163)$$

care stabilește o egalitate între valoarea optimă a funcționalei pentru cei N pași, suma variației funcției de performanță pentru $p-1$ pași și valoarea optimă a funcționalei pentru ceilalți $N-p$ pași.

În același mod se stabilește și formula de recurență inversă sub forma:

$$\Phi^{opt}_{N\to 1}(y_{N-1}) = F^{opt}_{CN}(x_{N-1}) + F^{opt}_{C(N-1)}(x_{N-2}) + \ldots + F^{opt}_{C(p-1)}(x_p) + \Phi^{opt}_{p\to 1}(y_{p-1}), \qquad (10-164)$$

care stabilește o egalitate între valoarea optimă a funcționalei pentru cei N pași, suma creșterii funcției de performanță pentru $N-p-1$ pași parcurși invers și valoarea optimă a funcționalei de performanță pentru ceilalți p pași.

Relațiile de recurență (10—163) și (10—164) mijlocesc determinarea valorii optime a funcționalei de performanță printr-o procedură de calcul în pași, prin care valoarea acesteia se exprimă numai prin variabila de comandă corespunzătoare pașilor parcurși, ceea ce constituie un mare avantaj al acestei metode.

Asupra variabilelor de stare și a variabilelor de comandă se pot impune restricții dependente de timp. Aceste restricții se pot introduce la anumite intervale de timp sau pași, dînd astfel posibilitatea cercetătorului să urmărească efectul lor.

Calculul valorii optime a funcționalei de performanță se execută fie după formula de recurență directă fie după formula de recurență inversă. După formula de recurență inversă procedura de calcul este următoarea:

— pentru pasul N și pentru fiecare pas înapoi se întocmește cîte un tabel cu valorile lui x_p, y_p corespunzătoare pasului p și valorile lui y_{p+1} corespunzătoare pasului $p+1$, deduse pe baza modelului procesului;

— se notează valorile funcției de performanță pentru toate valorile lui x_p și y_p pentru fiecare pas înapoi;

— se reține valoarea optimă a funcționalei de performanță obținută înaintea pasului curent, adică suma valorilor optime ale funcției de performanță pentru pașii parcurși înapoi;

— se adaugă valoarea funcției de performanță la valoarea optimă a funcționalei de performanță pentru pașii parcurși;

Scanned with OKEN Scanner

— se alege valoarea optimă a funcţionalei de performanţă pentru fiecare variabilă de conducere x_p;

— operaţiile menţionate se repetă pînă se atinge starea iniţială.

Optimizarea cu metoda programării dinamice este bogat ilustrată în literatura de specialitate. O prezentare mai amplă a acestei metode, însoţită de cîteva exemple de aplicaţie, se găsesc şi în lucrarea noastră [11] la paginile 145—212. Cititorul poate consulta în această direcţie lucrările [36], [35].

10.5. NOŢIUNI CU PRIVIRE LA PROIECTAREA ŞI IMPLEMENTAREA SISTEMELOR DE CONDUCERE CU CALCULATOARE

Obiective şi etape principale

Un sistem de conducere cu ajutorul calculatorului de proces (*SCC*) este un ansamblu complex constituit din:

— echipamentul de interfaţă proces-calculator;

— echipamentul de prelucrare a informaţiei („hardware");

— programe de exploatare şi de aplicaţie („software");

— personal de exploatare şi conducere,

organizat astfel încît să poată colecta şi prelucra în timp real un important flux de informaţie şi să execute comenzile ce se impun pentru a îndeplini anumite sarcini de producţie la nivelul unor performanţe măsurabile, dinainte stabilite.

Un astfel de sistem trebuie privit şi apreciat din punct de vedere istoric şi evolutiv în funcţie de condiţiile tehnice şi sociale în care se creează şi se dezvoltă.

În prezent, nivelurile şi ritmurile de dezvoltare a tehnicii şi tehnologiei conducerii automate variază foarte mult de la o ţară la alta şi de la un domeniu de aplicare la altul. Cauzele principale ale acestei situaţii sînt lipsa sau insuficienţa echipamentelor tehnice necesare, lipsa unor cadre bine pregătite în această specialitate, precum şi faptul că cel mai mare volum de muncă şi cea mai înaltă competenţă se cer a fi asigurate tocmai în etapa de început, etapa de analiză-proiectare-implementare a acestor sisteme.

Progresele mari înregistrate în ultimele două decenii în domeniul metodelor de analiză şi sinteză a sistemelor de conducere au dus la apariţia unei profesiuni noi: cea de analist de sistem şi la premisele abordării ştiinţifice a acestei probleme.

Realizarea fizică şi punerea în exploatare a unui sistem de conducere cu ajutorul calculatorului de proces se face, de regulă, în cîteva etape pe care le-am putea grupa astfel:

— elaborarea studiului de necesitate şi oportunitate;

— elaborarea studiului tehnico-economic;

— elaborarea proiectului de execuţie;

— realizarea (construirea) fizică a sistemului;

— testarea şi implementarea acestuia;

— operarea, întreţinerea şi perfecţionarea sistemului.

De regulă, trecerea de la o etapă la alta se face în urma unor analize şi aprobări ale forurilor de decizie, aşa cum se va arăta în cele ce urmează.

Scanned with OKEN Scanner

Studiul de necesitate și oportunitate (SNO). În cadrul acestui studiu se face o examinare generală a unității în care ar urma să se folosească un calculator de proces pentru a constata dacă unitatea respectivă se pretează la o conducere cu ajutorul calculatorului și dacă aceasta ar fi eficientă.

Trebuie să se plece de la principiul că un calculator poate fi util pentru conducerea oricărui proces, însă nu orice proces poate fi condus optim și economic de către orice calculator. De aceea, studiul trebuie să dea răspuns la următoarele întrebări:

— ce avem, ce dorim să facem, ce putem face?
— este necesară și oportună utilizarea unui calculator?
— ce tip de calculator și în ce mod să fie folosit?
— cît costă și care ar fi eficiența tehnico-economică.

Un studiu de oportunitate este oricînd util deoarece tehnica în acest domeniu evoluează foarte rapid și ceea ce astăzi nu este oportun, ar putea deveni oportun mîine.

Dacă, după lămurirea acestor probleme, factorii de decizie ajung la concluzia că este necesară și oportună utilizarea unui *SCC*, se redactează o temă de proiectare, apoi se trece la fazele următoare.

Studiul tehnico-economic (STE). Prin acesta se aprofundează examinarea începută în etapa anterioară pentru a stabili funcțiunile ce urmează a fi îndeplinite de către calculator. În acest scop echipa de specialiști însărcinată cu elaborarea studiului trebuie să analizeze cu multă atenție procesul tehnologic în cauză și să stabilească ce anume se poate și este oportun a fi condus cu calculatorul și cum?

Dacă este vorba de un proces tehnologic dintr-o instalație existentă trebuie să se examineze modificările și înlocuirile ce se impun, completările ce trebuie făcute pentru a asigura un flux rațional de informații și comenzi.

În cadrul unui *STE* trebuie să se fundamenteze apoi documentația privind alegerea configurației sistemului de calcul ce urmează a fi achiziționat pe baza analizei ofertelor firmelor furnizoare, cheltuielile și beneficiile scontate, resursele necesare, modul de organizare sau reorganizare a conducerii unității respective, precum și restructurarea serviciilor afectate de introducerea calculatorului.

În *STE* se precizează, de asemenea, spațiile necesare pentru sistemul de calcul adoptat, modalitatea de recrutare și formare a personalului de exploatare a acestuia, precum și posibilitatea de dezvoltare a sistemului inițial.

Dacă *STE* confirmă necesitatea, oportunitatea și eficiența sistemului propus el este aprobat de forurile competente care autorizează includerea lucrării în planul de investiții și elaborarea proiectului de execuție. În caz contrar se reface sau se abandonează.

Proiectul de execuție (PE). În cadrul acestui proiect se concretizează pînă în cele mai mici detalii prevederile formulate în etapele precedente.

Se elaborează sau se definitivează modelul sau modelele procesului condus prin procedeul identificare-modelare și eventual simulare a acestuia, împletind activitatea de proiectare cu cea de studiu și cercetare. Pe această bază se definitivează procedurile de culegere, transmitere, memorare, prelucrare și vizualizare a informațiilor, obiectivele de realizat, împărțirea răspunderilor între participanții la proiect, planificarea termenelor și a resurselor, metodele de evaluare a performanțelor sistemului precum și calculul eficienței tehnico-economice a sistemului propus.

Scanned with OKEN Scanner

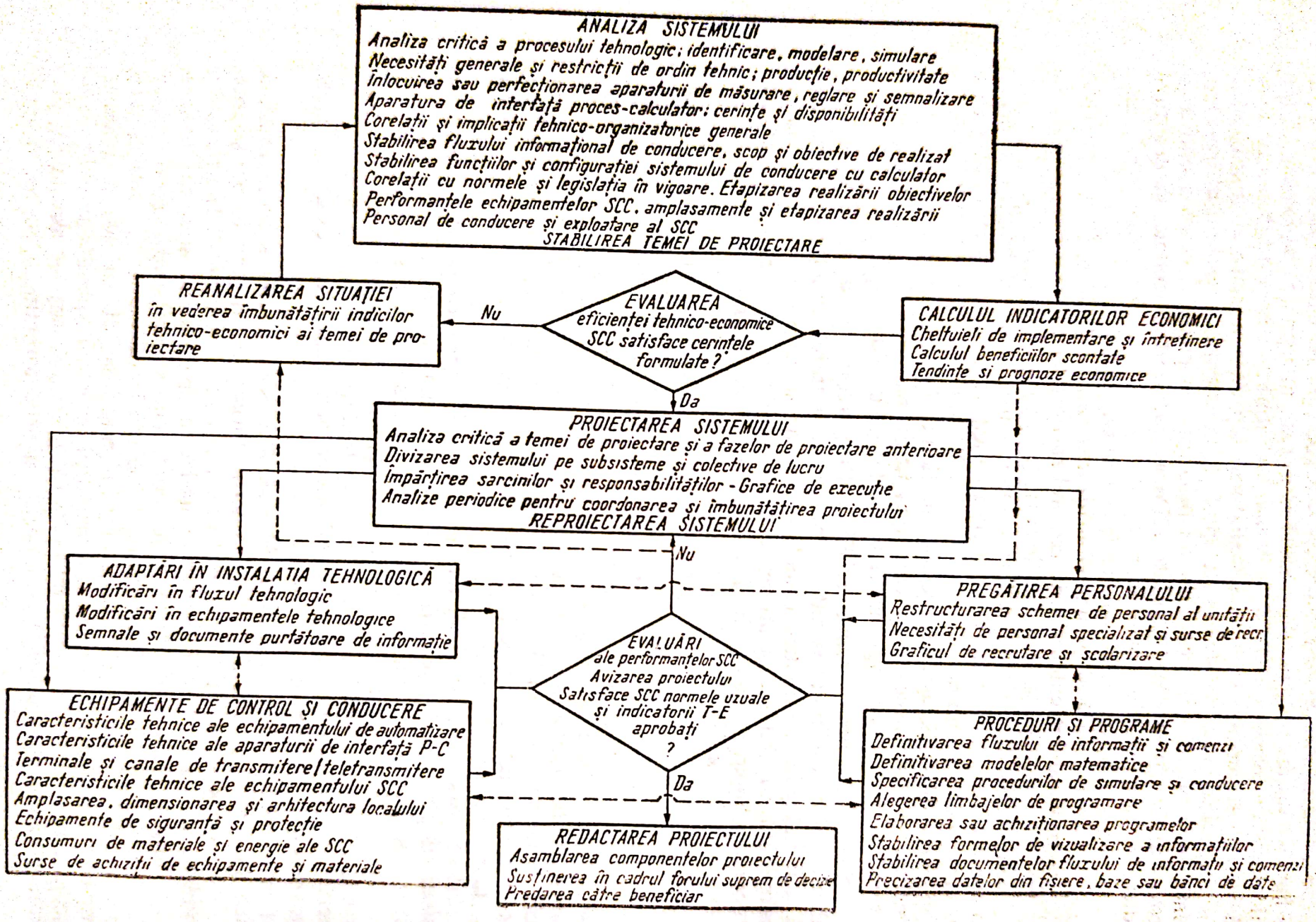


Fig. 10.60.

Scanned with OKEN Scanner

De asemenea, în această etapă se definitivează schemele logice ale programelor şi subprogramelor ce vor fi utilizate, se alege limbajul sau limbajele de programare, se redactează sau se achiziţionează programele şi subprogramele respective, se alcătuiesc colecţiile de date la care se va apela (fişiere, baze de date, bănci de date etc.), se fac noi precizări şi eventual noi opţiuni privind echipamentul necesar ş.a.

În forma sa finală, proiectul de execuţie reprezintă sinteza şi totodată concretizarea ideilor aprobate în etapele anterioare. De altfel, nici nu se poate stabili o graniţă certă între etapele menţionate mai înainte. Uneori *SNO* se poate include în *STE*; în alte cazuri *STE* şi *PE* s-ar putea executa într-o singură etapă.

În figura 10.60 prezentăm în sinteză un model de organigramă a activităţilor legate de proiectarea unui *SCC*.

Construcţia sistemului. Aceasta este etapa prin care se materializează în practică proiectul de execuţie elaborat în etapele anterioare. În această etapă se mai verifică încă odată documentaţia elaborată de proiectant, se montează şi se pune în stare de funcţionare echipamentul achiziţionat, se realizează legăturile fizice între diversele elemente componente ale sistemului de conducere.

Se verifică, de asemenea, programele şi subprogramele elaborate de proiectant şi se transpun pe suport fizic adecvat calculatorului achiziţionat, se alcătuieşte biblioteca de programe, se creează fişierele necesare ş.a.

Tot în această etapă se face prima compilare şi testare a programelor şi subprogramelor precum şi alte verificări care să permită depistarea eventualelor erori şi executarea corecţiilor ce se impun, apoi se pregăteşte documentaţia destinată utilizatorului prin care acesta va cunoaşte modul de funcţionare şi utilizare a sistemului de conducere în vederea preluării sale în exploatare.

Testarea şi implementarea sistemului. Este etapa în care se verifică performanţele pe care le oferă sistemul realizat şi dacă ele satisfac prevederile proiectului şi, în funcţie de rezultat, se trece la etapa de exploatare sau se reface proiectul.

Implementarea la o unitate deja existentă trebuie să se facă cu multă prudenţă întrucît în această etapă se trece de la vechiul sistem la noul sistem, existînd uneori chiar şi un al treilea sistem — un sistem tranzitoriu. Trecerea la noul sistem se poate face în paralel cu vechiul sistem, succesiv, pe secţiuni sau mixt. Pentru siguranţă se păstrează şi vechiul sistem sau se recurge la conducere directă de către personalul de conducere pînă cînd noul sistem funcţionează fără deficienţe.

Un sistem de conducere se consideră implementat în măsura în care acesta satisface performanţele scontate şi prezintă suficientă siguranţă în exploatare. Prin urmare nu se poate preciza o graniţă între etapa de construcţie şi cea de implementare a sistemului.

În partea finală a acestei etape se trece la reorganizarea conducerii unităţii respective, la unele restructurări ale compartimentelor afectate de introducerea calculatorului (aprovizionare, plan, contabilitate etc.) şi la predarea către beneficiar (utilizator) a documentaţiei privind utilizarea sistemului.

Scanned with OKEN Scanner

În figura 10.61 este prezentată organigrama activităţilor legate de construcţia şi implementarea unui *SCC*.

Operarea, întreţinerea şi perfecţionarea. Aceasta este etapa de producţie, de exploatare a sistemului implementat, care se caracterizează printr-o stabilitate (neschimbare) relativă. În această etapă se lichidează eventualele deficienţe semnalate în timpul exploatării sistemului căruia i se aduc treptat noi îmbunătăţiri atît prin ridicarea performanţelor iniţiale cît şi prin realizarea de noi funcţii.

Implicaţiile implementării unui SCC

Realizarea unui sistem de conducere cu calculator *(SCC)* este o problemă deosebit de complicată. Ea prezintă o serie de caracteristici şi particularităţi de care trebuie să se ţină seama de la început. Dintre acestea remarcăm următoarele:

a) Implementarea unui astfel de sistem are implicaţii tehnico-economice, tehnologice şi sociale deosebite, iar reuşita acţiunii, eficienţa acesteia depind de competenţa şi conştiinciozitatea personalului însărcinat cu această misiune.

Dacă implementarea se face pe un proces şi o instalaţie tehnologică nouă, în curs de proiectare, atunci există posibilitatea îmbunătăţirii substanţiale, revoluţionare, nu numai a conducerii procesului respectiv, ci şi a revoluţionării procesului tehnologic însăşi, bazată pe imensele posibilităţi ale calculatorului de a colecta şi prelucra informaţia tehnologică necesară şi de a efectua comenzile ce se impun. De exemplu, într-un mod este concepută o linie tehnologică deservită manual şi în cu totul alt mod este concepută o linie tehnologică în întregime automatizată, comandată de un calculator de proces.

Implementarea nu trebuie să se limiteze la îndeplinirea de către calculator a vechilor metodologii şi tehnici de conducere, a vechilor obiceiuri şi rutine, ci este necesar ca acestea să fie revăzute şi îmbunătăţite substanţial, conceptual pe măsura marilor posibilităţi oferite de calculator. A utiliza un calculator de proces înseamnă a accepta o revoluţionare.

b) Implementarea şi integrarea completă a unui astfel de sistem poate dura timp îndelungat şi necesită cheltuieli importante pînă la $20 \div 30\%$ din costul instalaţiei tehnologice.

Perioada de studii, analize şi proiectare poate dura luni şi chiar ani de zile în funcţie de complexitatea procesului tehnologic în cauză şi este greu să se spună cînd se încheie o astfel de activitate caracterizată prin frecvente modificări şi îmbunătăţiri, care se aduc mereu pe măsură ce se cunoaşte mai bine situaţia, chiar şi în faza de construcţie, testare şi implementare a sistemului.

Etapei de proiectare trebuie să i se acorde cea mai mare importanţă deoarece ea prezintă cele mai mari riscuri de a începe şi realiza un proiect necorespunzător. De aceea, la această muncă de studii, analiză şi proiectare trebuie să se atragă cel mai talentat şi creator personal din întreprinderea beneficiară a sistemului şi din cea constructoare a acestuia.

Implementarea presupune cheltuieli importante greu de stabilit de la început, mai ales dacă utilizarea calculatorului se face pentru prima dată în domeniul respectiv. De asemenea, în cele mai multe cazuri nu se cunoaşte cu precizie de la început ceea ce trebuie făcut, întrucît

Scanned with OKEN Scanner

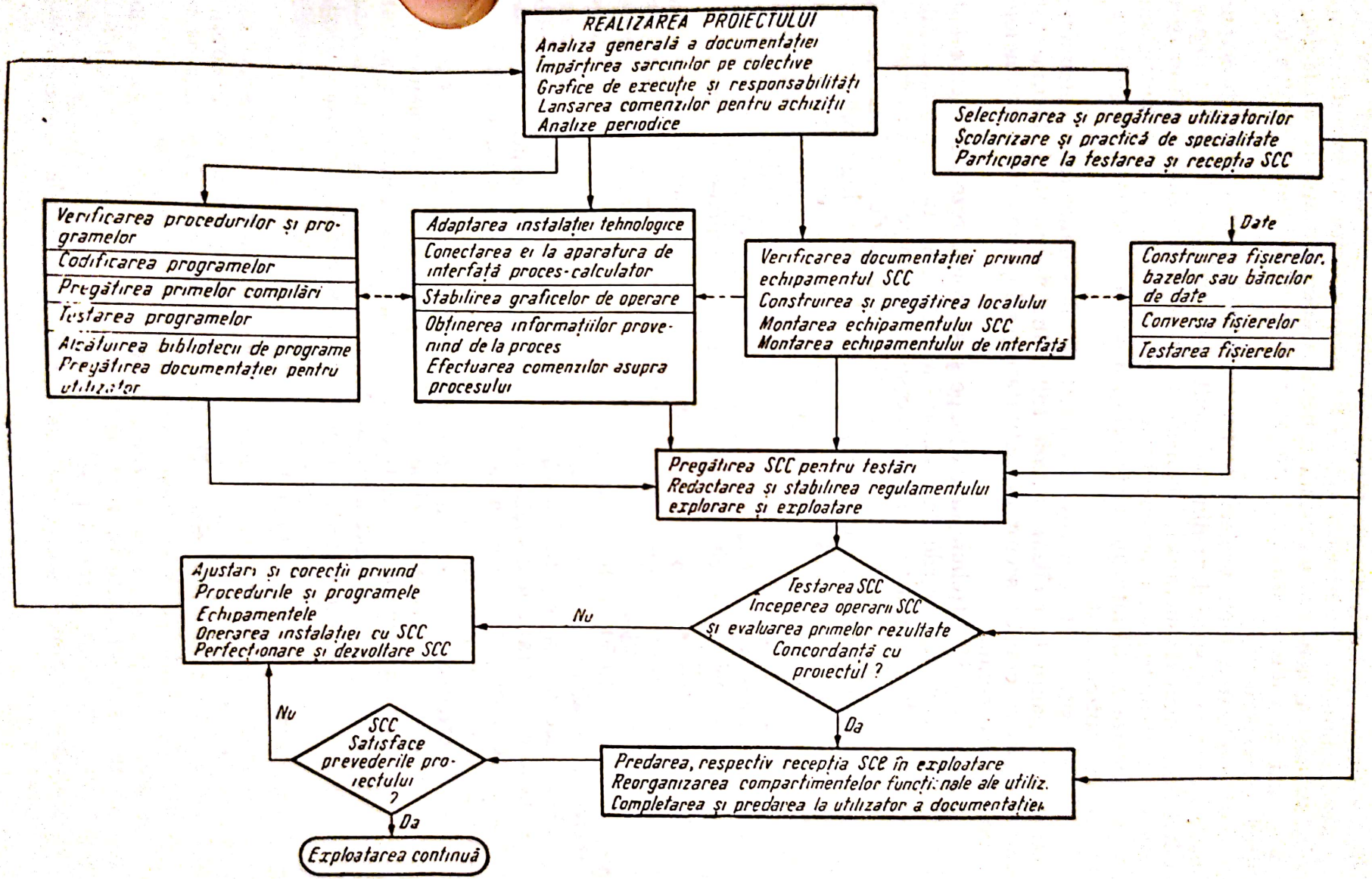

REALIZAREA PROIECTULUI
Analiza generală a documentației
Împărțirea sarcinilor pe colective
Grafice de execuție și responsabilități
Lansarea comenzilor pentru achiziții
Analize periodice
Selecționarea și pregătirea utilizatorilor
Școlarizare și practică de specialitate
Participare la testarea și recepția SCC
Verificarea procedurilor și programelor
Codificarea programelor
Pregătirea primelor compilări
Testarea programelor
Atașarea bibliotecii de programe
Pregătirea documentației pentru utilizator
Adaptarea instalației tehnologice
Conectarea ei la aparatura de interfață proces-calculator
Stabilirea graficelor de operare
Obținerea informațiilor provenind de la proces
Efectuarea comenzilor asupra procesului
Verificarea documentației privind echipamentul SCC
Construirea și pregătirea localului
Montarea echipamentului SCC
Montarea echipamentului de interfață
Date
Construirea fișierelor, bazelor sau băncilor de date
Conversia fișierelor
Testarea fișierelor
Pregătirea SCC pentru testări
Redactarea și stabilirea regulamentului explorare și exploatare
Ajustări și corecții privind
Procedurile și programele
Echipamentele
Operarea instalației cu SCC
Perfecționare și dezvoltare SCC
Testarea SCC
Începerea operării SCC
și evaluarea primelor rezultate
Concordantă cu proiectul ?
Nu
Da
SCC
Satisface prevederile proiectului ?
Nu
Da
Predarea, respectiv recepția SCC în exploatare
Reorganizarea compartimentelor funcționale ale utiliz.
Completarea și predarea la utilizator a documentației
Exploatarea continuă

Fig. 10.61.

fiecare sistem de conducere cu calculator este un unicat. Unele obiective şi sarcini propuse iniţial urmează să fie reconsiderate pe măsură ce studiul avansează, deoarece nu există suficientă documentaţie în acest domeniu, mai ales din cauza păstrării secretului profesional de către unele firme cu experienţă.

c) Realizarea unui proiect de implementare a unui *SCC* presupune participarea unor specialişti din mai multe domenii de activitate: ingineri tehnologi, ingineri mecanici, ingineri electronişti, ingineri automatişti, analişti, matematicieni, economişti ş.a. precum şi participarea nemijlocită şi permanentă a utilizatorului sistemului, adică a beneficiarului procesului tehnologic automatizat.

Această situaţie face necesară o foarte bună conducere şi comunicare între participanţii la realizarea proiectului, întocmirea unei documentaţii clare pe fiecare etapă şi sector de activitate şi adoptarea unei terminologii, a unui limbaj comun.

Beneficiarul *SCC* trebuie să participe şi să conlucreze activ cu ceilalţi participanţi la toate etapele de realizare ale *SCC*, să furnizeze analiştilor de sistem datele necesare despre procesul tehnologic ce urmează a fi automatizat, să fie de acord cu propunerile acestora de a aduce îmbunătăţiri nu numai conducerii ci şi însăşi procesului tehnologic automatizat, să asigure din timp pregătirea personalului necesar exploatării noului sistem ş.a. Prin urmare, beneficiarul unui *SCC* trebuie să revizuiască ordinea existentă în unitatea sa, să elimine tot ce este perimat şi nerațional, să descopere şi să folosească rezervele ascunse, să introducă o nouă ordine care să asigure o cît mai înaltă eficienţă tehnico-economică.

d) Realizarea unui proiect de implementare a unui *SCC* este un proces iterativ care presupune plecarea de la general spre detaliu şi reconsiderarea ideilor iniţiale pe măsura avansării proiectului, aşa cum reiese şi din figurile 10.50 şi 10.51 unde se sugerează aceasta prin legăturile inverse de evaluare şi corecţie ale prevederilor iniţiale.

Este necesar să se stabilească în mod corect criteriile de evaluare, să se ia în consideraţie mai multe alternative, mai multe soluţii, iar avansarea să se facă pas cu pas în deplină cunoştinţă de cauză, evitîndu-se salturile mari care sînt riscante.

e) Efectuarea unui studiu în vederea implementării unui *SCC* are întotdeauna efecte favorabile chiar şi cînd nu se trece la implementare şi cu atît mai mult după ce se trece la această fază. Practica a arătat că analiza critică şi riguroasă a proceselor tehnologice reuşeşte să scoată la iveală rezerve încă nefolosite în privinţa îmbunătăţirii acestora. Deci, astfel de studii sînt utile prin ele însele, întrucît duc la o mai bună înţelegere şi stăpînire a procesului tehnologic în cauză.

Ameliorarea indicatorilor tehnico-economici (producţie, productivitate, preţ de cost, calitate ş.a.) cu numai 1—2% compensează cu prisosinţă cheltuielile făcute în acest scop. De altfel trebuie avut în vedere că în afară de efectele tehnico-economice evaluabile în valori băneşti sînt şi efecte greu sau imposibil de evaluat şi pot fi cazuri cînd acestea din urmă să fie preponderente în argumentarea folosirii unui *SCC*; de pildă, asigurarea securităţii oamenilor şi a instalaţiilor, satisfacerea gusturilor consumatorilor, schimbarea caracterului muncii personalului de exploatare şi conducere ş.a.

Scanned with OKEN Scanner

Capitolul 11

SISTEME DE ECHIPAMENTE

Ca orice sistem de calcul automat (care funcţionează cu programe memorate) şi calculatoarele de proces au o structură complexă formată din două părţi funcţionale complet distincte şi anume: *sistemul de echipamente* (aşa-numita parte de „hard") şi *sistemul de programe* (partea de „soft").* La generaţia a IV-a de echipamente se consideră ca distinct şi *sistemul de microprograme* (denumit şi „firmware"); microprogramele (care sînt introduse în dispozitive de memorare speciale, inaccesibile programelor utilizatorilor) au rolul de a simula unele instrucţiuni speciale cu ajutorul instrucţiunilor elementare din setul de bază al calculatorului.

În cadrul acestui capitol vom face o prezentare numai a sistemului de echipamente, cu o dezvoltare la nivelul necesar beneficiarilor calculatoarelor de proces (adică a inginerilor tehnologi), insistînd totuşi asupra particularităţilor specifice acestor calculatoare specializate, precum şi a dispozitivelor din structura lor generală care nu sînt întîlnite în mod uzual în configuraţia calculatoarelor numerice universale (ca de exemplu: cuplorul de proces, canalele de intrare/ieşire cu interfeţe caracteristice mai multor nivele de transmitere a informaţiei etc.).

11.1. CONCEPŢIA GENERALĂ A CALCULATOARELOR DE PROCES

Aşa cum am reţinut şi din subcapitolul 10.2, calculatoarele de proces sînt calculatoare specializate cu o configuraţie extrem de diversă, determinată atît de modul de utilizare a calculatorului pentru conducerea procesului cît şi de natura şi particularităţile procesului însuşi. În principiu, calculatoarele de proces sînt calculatoare electronice numerice universale prevăzute cu opţiuni specifice (printre care cea mai importantă este interfaţa industrială sau de proces), care le permit să interacţioneze direct cu procesul tehnologic condus (fig. 11.1). Să reţinem că în unele cazuri mai deosebite (cele mai multe din domeniul petrochimiei şi transportului gazelor naturale), calculatorului numeric universal este indicat să i se

* Ansamblul de echipamente şi programe utilizate în conducerea proceselor a fost denumit în literatura tehnică SICCPTEH (acronim de la „SIstem de Conducere cu Calculatoare a Proceselor TEHnologice").

Scanned with OKEN Scanner

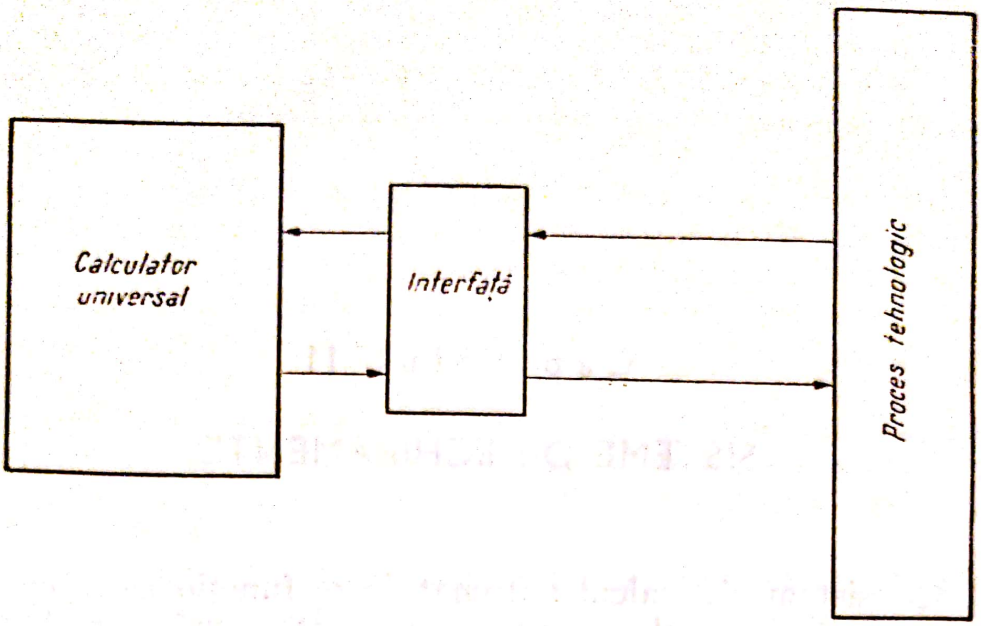


Fig. 11.1.

asocieze şi un calculator analogic universal; dacă cele două calculatoare universale, numeric şi analogic, sînt interconectate între ele (printr-o interfaţă hibridă), atunci calculatorul de proces este un calculator hibrid (acesta fiind, evident, cazul general). După cum se ştie, o interconectare hibridă creează condiţiile favorabile pentru combinarea avantajoasă a proprietăţilor complementare ale calculatoarelor analogice (acestea, ca „maşini-paralel", sînt foarte indicate la integrarea ecuaţiilor diferenţiale sau cu derivate parţiale, prin care sînt modelate procesele tehnologice cu parametri distribuiţi) şi ale calculatoarelor numerice (cu mari resurse privind: versatilitatea, independenţa de „drift", memorarea şi deciziile logice).

Configuraţia calculatoarelor de proces

În subcapitolul 10.2 a fost prezentată *arhitectura* unui calculator de proces, adică principiile fundamentale ale structurii şi modului de funcţionare (referitoare la: echipamentele de bază, modalităţi de interconectare, transmiterea datelor, organizarea generală, priorităţile, întreruperile etc.). Aici vom da cîteva detalii cu privire la *configuraţia* sistemelor de conducere a proceselor cu ajutorul calculatoarelor numerice automate.

După cum se ştie, prin configuraţia unui sistem de calcul se înţelege un anumit „set" de unităţi de echipamente reunite printr-o schemă funcţională. Putem vorbi de o *configuraţie generală* (principială, adecvată unor clase de aplicaţii), de o *configuraţie standard* (specifică unor anumite tipuri de sisteme realizate de firmele constructoare pentru un anume gen de procese) sau de o *configuraţie efectivă* (realizată numai într-un anume scop, pentru o aplicaţie dată, cu elemente specifice ei). De exemplu, putem analiza configuraţia generală a unui sistem de conducere cu calculatoare automate universale a proceselor tehnologice industriale sau configuraţia generală a unui multisistem de calculatoare pentru asigurarea conducerii la mai multe nivele ierarhice. Tot aşa, se poate prezenta configuraţia standard a unui calculator de proces pentru culegerea şi interpretarea date-

Scanned with OKEN Scanner

lor fizice sau configurația standard a calculatoarelor de proces: FELIX C—32P, IBM System/7, IBM 1800 [27] cu aplicabilitate în petrochimie, T 2000 pentru aplicații de telemecanică [32], sau configurația standard a unui sistem „on-line" de control permanent și evidență a producției într-o întreprindere de construcții mecanice, folosind echipamentele IBM 370 (ca sistem de calcul central), IBM System/7 (pentru culegerea datelor și IBM 2790 (ca sistem de comunicație). O realizare practică, într-un caz particular (de exemplu, utilizarea calculatorului de proces FELIX C—32P, pentru analiza proceselor tehnologice dintr-o linie de elaborare a cimentului de la întreprinderea *x*) prezintă o configurație efectivă, specifică numai acestei aplicații, însă în concordanță cu configurația generală și cu configurația standard posibilă.

Configurația generală a unui calculator de proces destinat întreprinderilor industriale este prezentată în figura 11.2, unde au fost puse în evidență numai acele părți structurale de bază specifice funcției (destinației) de conducere automată a proceselor industriale. Notațiile folosite în schema din figura 11.2 au următoarele semnificații: *UCP* — unitatea centrală de prelucrare (cu: *MI* — memoria internă; *BA/L* — blocul pentru operațiile aritmetice și logice: *BC* — blocul de comandă; *T* — „timer", adică un generator de tact adecvat funcționării în timp real); *UPS* — unități periferice standard (pentru memoria externă cu discuri magnetice, benzi magnetice etc., cititoare/perforatoare de cartele/bandă de hîrtie, imprimante rapide, unități de afișare cu tub catodic de tip „display", trasoare de grafice de tip „incremental digital plotter", consolă etc.); *POT* — pupitrul operatorului tehnolog; *SI/E* — sistemul de introducere/extragere a datelor (cu: *LPr.* — legături programate și *LN* — legături neprogramate prin: *CM* — canal multiplexor și *CS* — canale selectoare); *UC* — unitate de cuplare („cuplor"); *MIA* — module pentru intrările analogice; *MIN* — module pentru intrările numerice; *MEA* — module pentru ieșirile analogice; *MEN* — module pentru ieșirile numerice*; *TA* — traductoare de proces („senzori") de tip analogic; *TN* — traductoare de proces („senzori") de tip numeric (digital); *EA* — dispozitive periferice pentru reglarea procesului („executori") de tip analogic și *EN* — dispozitive periferice pentru reglarea procesului („executori") de tip numeric.

Așa cum rezultă din configurația generală (fig. 11.2), calculatoarele de proces au o structură modulară, ceea ce permite adaptarea lor la diversele aplicații concrete ale utilizatorului, prin „asamblarea" unui anumit număr de elemente structurale și realizarea de configurații diverse.

Din figura 11.2 rezultă că elementele de bază ale unui sistem de conducere automată a proceselor cu ajutorul tehnicilor de calcul numerice sînt: unitatea de prelucrare, unitățile de intrare/ieșire, interfețele pentru mai multe nivele de conexiuni și perifericele de proces (care sînt specifice naturii procesului tehnologic controlat).

Unitățile de prelucrare pot diferi prin: viteza de calcul, capacitatea memoriei interne și posibilitățile utilizării memoriei externe (unități cu

* Aceste module ale interfeței industriale, precum și unitatea de cuplare, vor fi prezentate mai în detaliu în cadrul subcapitolului 11.3 — referitor la sistemele de intrare/ieșire.

Scanned with OKEN Scanner

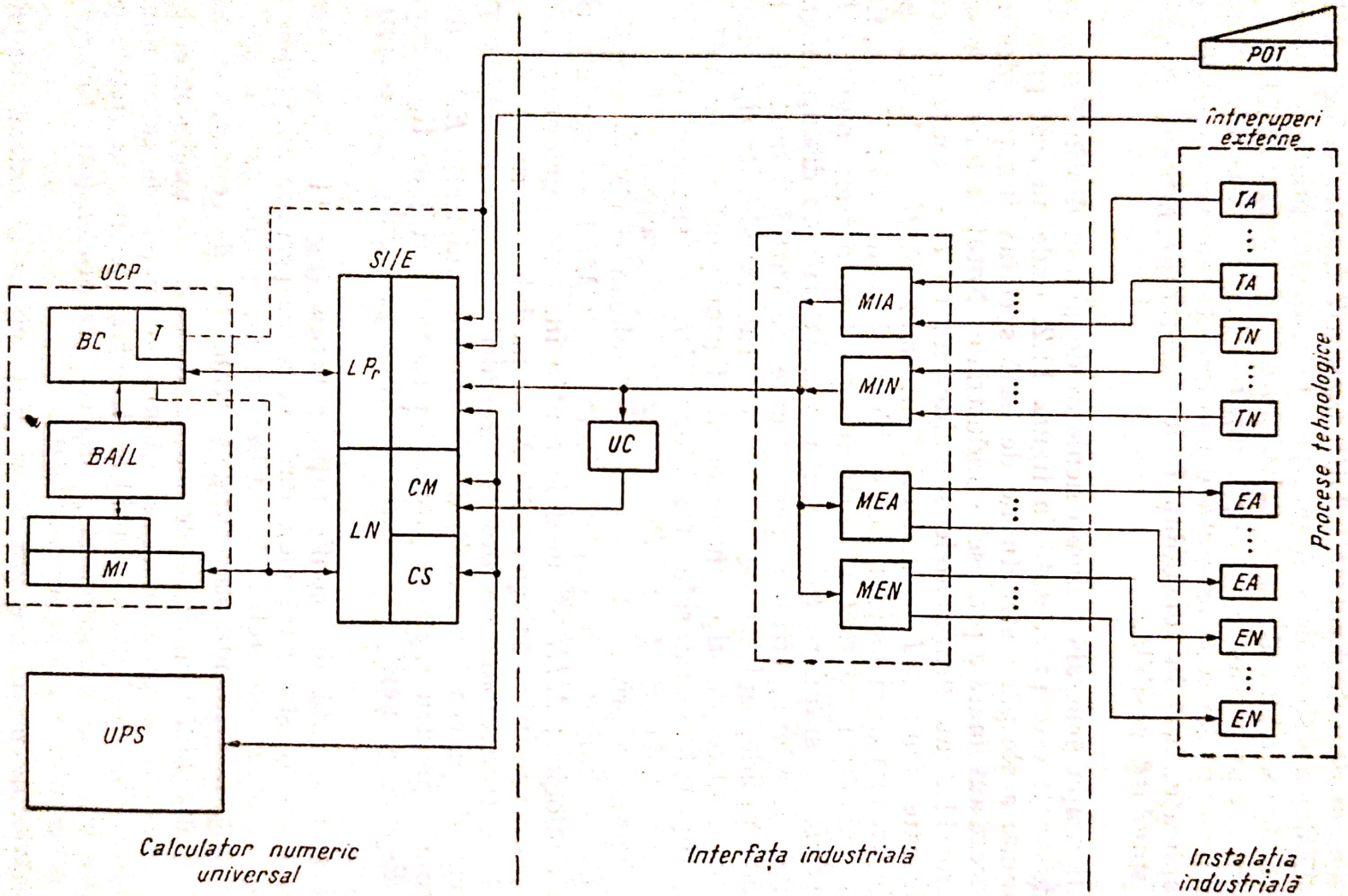


Fig. 11.2.

Scanned with OKEN Scanner

benzi şi/sau discuri magnetice). Oricum, unitatea centrală de prelucrare trebuie să posede un set de instrucţiuni de bază pentru: operaţiile aritmetice în „simplă" şi „dublă" precizie, operaţiile logice şi operaţiile de intrare-ieşire. Memoria internă este formată din module (v. fig. 11.2), mărimea modulului de bază fiind — în general — de 5 096 ($=2^{12}$) cuvinte, iar cuvîntul-memorie avînd lungimea de 16 bits informaţionali (adică 2 octeţi sau 2 Bytes) plus 2 bits de control.

Unităţile de intrare-ieşire sînt foarte diverse: cititoare de cartele, unităţi de bandă magnetică, imprimante, trasoare de grafice ş.a. Mai mult, prin intermediul unor unităţi de legătură, calculatorul de proces poate fi „conectat" cu un calculator de gestiune (v. *Ex. 11—2*) sau/şi cu un calculator analogic (specializat pentru anumite modele care intervin frecvent în conducerea unui proces); în acest fel, poate rezulta un sistem complex, dar omogen şi „deplin", care realizează: conducerea automată a proceselor, calcule tehnice-ştiinţifice şi lucrări de gestiune, regăsirea informaţiei etc.

Aşa cum am mai arătat (v. fig. 10.20), se pot realiza scheme prin care mai multe calculatoare de proces sînt supervizate de un calculator puternic de gestiune, realizînd conducerea unui întreg ansamblu industrial (v. *Ex. 11—13*).

Calculatoarele de proces trebuie să fie astfel concepute încît să poată lucra prin „divizarea timpului" pentru executarea aparent simultană a mai multor lucrări, ale căror programe sînt, toate, prezente în acelaşi timp în memoria internă; în acest caz, un program aflat în lucru poate fi întrerupt oricînd în favoarea unui alt program cu prioritate mai mare (multiprogramare). Acest mod de lucru ţine nu numai de sistemul de operare („software"), dar şi de echipament (tehnologie de construcţie, dotare, capacitatea memoriei interne ş.a.).

Aşa cum rezultă din schema configuraţiei generale (v. fig. 11.2), schimbul de date între unităţile de intrare/ieşire, inclusiv perifericele de proces, se face prin intermediul unor „canale" de comunicaţie (ce fac obiectul subcapitolului 11.3), care asigură simultaneitatea transmisiilor.

Un rol important îl au, la calculatoarele de proces, şi întreruperile externe (v. fig. 11.2) realizate de dispozitivele de control ale procesului industrial; asupra lor vom reveni în paragraful imediat următor.

Exemple de configuraţii standard (tipice unor firme sau aplicaţii) şi de configuraţii efective (ale unor calculatoare de proces deja implementate) sînt foarte numeroase. În continuare, vom prezenta numai cîteva cazuri mai ilustrative, în special pe cele legate de posibilităţile de utilizare imediată în ţara noastră.

Ex. 11—1: configuraţia calculatorului de proces **FELIX C—32P.** Sistemul FELIX C—32P este primul calculator de proces de concepţie românească din generaţia a 3-a (realizare: Institutul de cercetări şi proiectări pentru tehnică de calcul — ITC şi Institutul de cercetări şi proiectări pentru automatizări — IPA) ce se construieşte în producţie de serie.

În figura 11.3 este prezentată schema configuraţiei standard a calculatorului de proces FELIX C—32P, în care: *MS* — maşină de scris; *CP* — cititor de cartele perforate; *IR* — imprimantă rapidă; *DM1* şi *DM2* — unităţi de discuri magnetice; *ULS* — unităţi de legătură simplă; *ULM* — unitate de legătură multiplă; *MI* — memorie internă, cu două

Scanned with OKEN Scanner

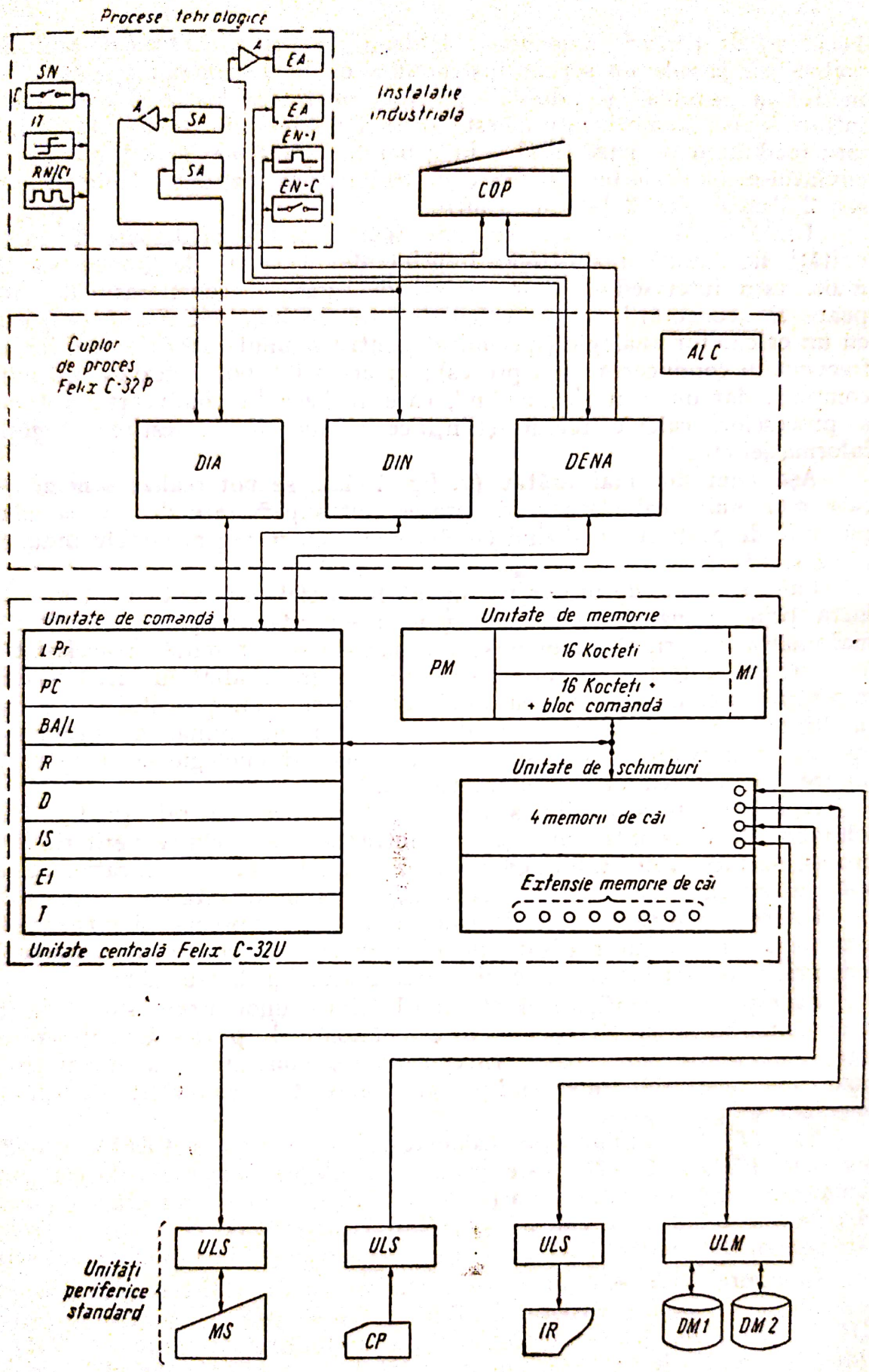


Fig. 11.3.

Scanned with OKEN Scanner

module de cîte 16 384 octeţi (adică cu o capacitate totală de 16 384 cuvinte de cîte 16 bits, echivalentă cu 32 Kocteţi), care poate fi — opţional — majorată la patru module a 16 Kocteţi fiecare; *PM* — bloc de protecţie a memoriei; *LPr* — dispozitiv pentru legăturile programate; *PC* — panou de comandă; *BA/L* — blocul pentru operaţiile aritmetice şi logice; *R* — registrele generale şi de stare-program; *D* — blocul procedurilor pentru derutări; *IS* — blocul pentru întreruperile standard; *EI* — blocul pentru extensia procedurilor de întreruperi; *T* — „timer" (orologiu de numărare); *DIA* — dispozitiv pentru intrările de proces analogice; *DIN* — dispozitiv pentru intrările de proces numerice; *DENA* — dispozitiv pentru ieşirile „la proces" numerice şi analogice; *AlC* — sursă de alimentare cu energie electrică a dispozitivelor cuplorului; *A* — amplificatoare de tensiuni electrice; *SN* — senzori (traductoare) de tip numeric; *SA* — senzori (traductoare) de tip analogic; *C* — traductoare cu contacte; *IT* — traductoare cu mărimea de ieşire sub formă de pulsuri („trepte") de tensiune electrică; *RN/CI* — registre numerice sau/şi contoare de pulsuri; *EA* — elemente de ieşire („executori") de tip analogic; *EN—I:* elemente de ieşire („executori") de tip numeric cu semnale în formă de pulsuri; *EN—C:* elemente de ieşire („executori") de tip numeric cu contacte şi *COP* — consola operatorului de proces.

După cum rezultă din această schemă (fig. 11.3) sistemul este format, în principal, din următoarele grupe de echipamente: un calculator electronic universal FELIX C—32U (cu „periferia" sa standard), un cuplor de proces FELIX C—32P şi un grup divers de periferice de proces*.

Specific tipului de aplicaţii de care ne ocupăm este, evident, aşanumitul *cuplor de proces* (care va fi prezentat, mai în detaliu, în cadrul exemplului *Ex. 11—9*). Acest cuplor a fost astfel conceput, ca o interfaţă industrială, încît să permită utilizarea calculatorului universal de capacitate mică FELIX C—32U la conducerea unor procese tehnologice din diverse ramuri industriale ca, de exemplu: chimie, metalurgie, materiale de construcţii etc. În configuraţia din figura 11.3, sistemul FELIX C—32P poate îndeplini următoarele funcţiuni: colectarea de date (valori ale mărimilor de stare chimico-fizice, specifice proceselor) şi supravegherea centralizată; regimul de conducere a proceselor de gen „ghid operator" şi comanda mărimilor de referinţă ale regulatoarelor utilizate pentru automatizarea procesului. Ca urmare, cuplorul poate asigura intrarea/ieşirea datelor prin semnale de forma: numerice (de tip contact, variaţii „treaptă" a tensiunilor electrice şi trenuri de pulsuri de tensiuni electrice) şi analogice (de tensiune sau de curent), semnale ce pot rezulta de la traductoarele dispozitivelor de automatizare convenţionale (care devin, astfel, dispozitive periferice de proces — „senzori").

Cuplorul de proces FELIX C—32P prezintă o structură modulară, asigurîndu-se astfel realizarea unui ansamblu flexibil (versatil); în funcţie de tipul, numărul şi capacitatea dispozitivelor specializate cuplate la calculatorul FELIX C—32U, ansamblul se poate realiza în diverse configuraţii efective. Astfel, se pot cupla pe lîngă opţiunile *COP* pînă la 10 dispozitive *DIN*, *DIA* şi *DENA* (v. fig. 11.3), în orice combinaţie.

---

* Bineînţeles că acestor echipamente le sînt ataşate şi sistemele de programe de bază: „general SFC-32" pentru calculatorul universal FELIX C-32U şi „de proces PROSIS-32" pentru cuplor (v. cap. 12, *Ex. 12—1*).

Scanned with OKEN Scanner

Datorită destinației de supraveghere și conducere a proceselor tehnologice s-a impus necesitatea lucrului în timp real și — de aici — a rezultat nevoia de conectare a cuplorului la calculatorul FELIX C—32U prin intermediul legăturii programate (v. fig. 11.3). Schimbul de date între proces și unitatea centrală de prelucrare se face sub controlul programului, cu ajutorul instrucțiunilor de citire și scriere directă (*WD* și *RD*, prezentate prin exemplul *Ex. 11—4*). Totuși, dintre tipurile de dispozitive specializate ce intră în componența cuplorului de proces, numai consola operatorului de proces (*COP* — v. fig. 11.3) nu se conectează direct la legătura programată; schimbul de date între *COP* și unitatea centrală este asigurat de interfața consolei operatorului de proces: cu *DIN* (v.fig. 11.3) — pentru introducerea de date în calculator și cu *DENA* — pentru extragerea de date din calculator.

Ca realizare constructivă, dispozitivele specializate *DIN*, *DIA* și *DENA* (v. fig. 11.3) sînt montate într-un dulap standard (tip FELIX C—32). În același dulap, este instalat și alimentatorul cuplorului (*AlC* pe fig. 11.3), care are o structură modulară ce cuprinde: un modul de alimentare (format din două redresoare de 30V/30A, elemente de protecție, disjuncție și alimentări auxiliare); un stabilizator de tensiune (pentru alimentarea circuitelor logice) și o sursă auxiliară de tensiune necesară alimentării unor circuite din dispozitivele specializate *DIN*, *DIA* și *DENA*.

Consola operatorului de proces (*COP* pe fig. 11.3) a fost realizată ca o unitate independentă. Peretele ei frontal are două panouri distincte: unul pentru afișările cu tuburi Nixie și lămpi de semnalizare (pentru informarea operatorului asupra desfășurării procesului) și altul cu elementele de comandă ce permit introducerea unor modificări în cadrul programului de conducere a procesului.

Ex. 11—2: configurația calculatorului de proces **System/7.** Calculatorul System/7 (S/7), produs de firma IBM, este destinat colectării de date referitoare la parametrii fizici ai procesului industrial condus, obținute prin măsurători directe de către traductoare, aparatele de măsurat sau alte elemente de supraveghere ale procesului (din structura dispozitivelor convenționale de automatizare). Semnalele electrice (de formă analogică sau discretă-numerică), după ce sînt transformate în modulul de prelucrare al calculatorului IBM System/7 în date (valori) semnificative, sînt înregistrate și apoi prelucrate. După prelucrare, calculatorul IBM S/7 poate elabora semnale de comandă pentru reglarea procesului; aceste semnale de ieșire din calculator realizează o reglare permanentă, în timp real, a procesului, pe întreaga durată de supraveghere. În același timp, IBM S/7 poate comunica cu o așa-numită stație de operare (fie pentru primirea unor programe de prelucrare, date de control etc., fie pentru comunicarea de mesaje și valori cu privire la procesul supravegheat) și, totodată, poate realiza înregistrări de date și programe printr-o unitate proprie de memorie cu discuri magnetice. De asemenea, IBM S/7 poate fi conectat și cu un calculator principal.

IBM System/7 are o structură pronunțat modularizată, putînd fi realizat în patru configurații standard (v. fig. 11.6). În oricare din aceste configurații, IBM S/7 se compune, în esență, dintr-o *unitate centrală* (formată dintr-un modul de prelucrare și 1 pînă la 11 module de intrare/ieșire) și o *stație de operare*, care — din punct de vedere constructiv—

Scanned with OKEN Scanner

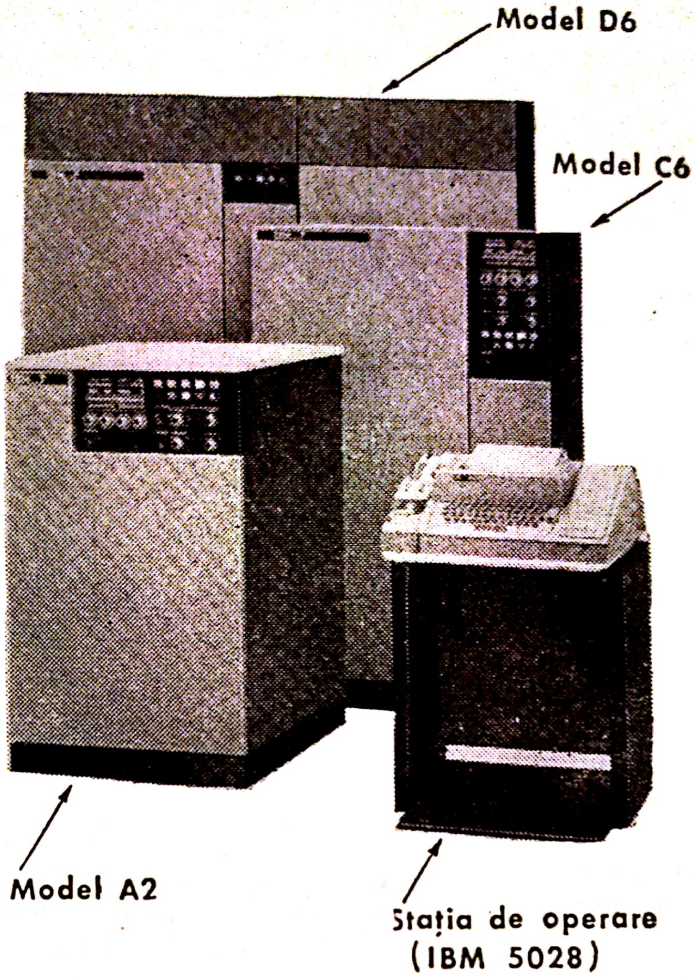


Fig. 11.4.

reprezintă echipamente distincte. Unitatea centrală poate fi amplasată într-un singur dulap (realizat în două variante constructive: modelul A2 și modelul C6, acesta putînd fi cu sau fără extensie pentru module de intrare/ieșire) sau în două dulapuri distincte (unul reprezentînd modelul C6 cu extensie, cuplat cu modelul D6 — un dulap separat conținînd numai module de intrare/ieșire). Aspectul exterior al acestor echipamente ale calculatorului de proces IBM System/7 este redat în figura 11.4.

Configurația standard de bază a calculatorului de proces IBM System/7 (referitoare la modelul A2) este redată în schema din figura 11.5, în care: *CO* reprezintă consola operatorului de la calculator; *MP* — memoria principală; *C—I/E* — dispozitivul de control pentru intrare/ieșire al modulului de prelucrare; *T* — „timer" (generatorul de tact pentru intervalele de timp); *MBS* — modul de bază pentru senzori, adică o interfață de conectare la perifericele de proces (senzori, dispozitive de control etc.) cu: *DIA* — dispozitive pentru intrările analogice *(IA)* și *DM* — dispozitive multifuncționale pentru intrările și ieșirile analogice și numerice *(I/E—A/N)*; *MMD* — modul de memorie pe discuri magnetice; *I/E—HB* — intrare/ieșire prin bandă de hîrtie (cititor bandă de hîrtie perforată și perforator bandă de hîrtie); *E—IMP* — ieșire prin imprimantă (mașină de scris); *I—CL* — intrare prin claviatură.

Scanned with OKEN Scanner

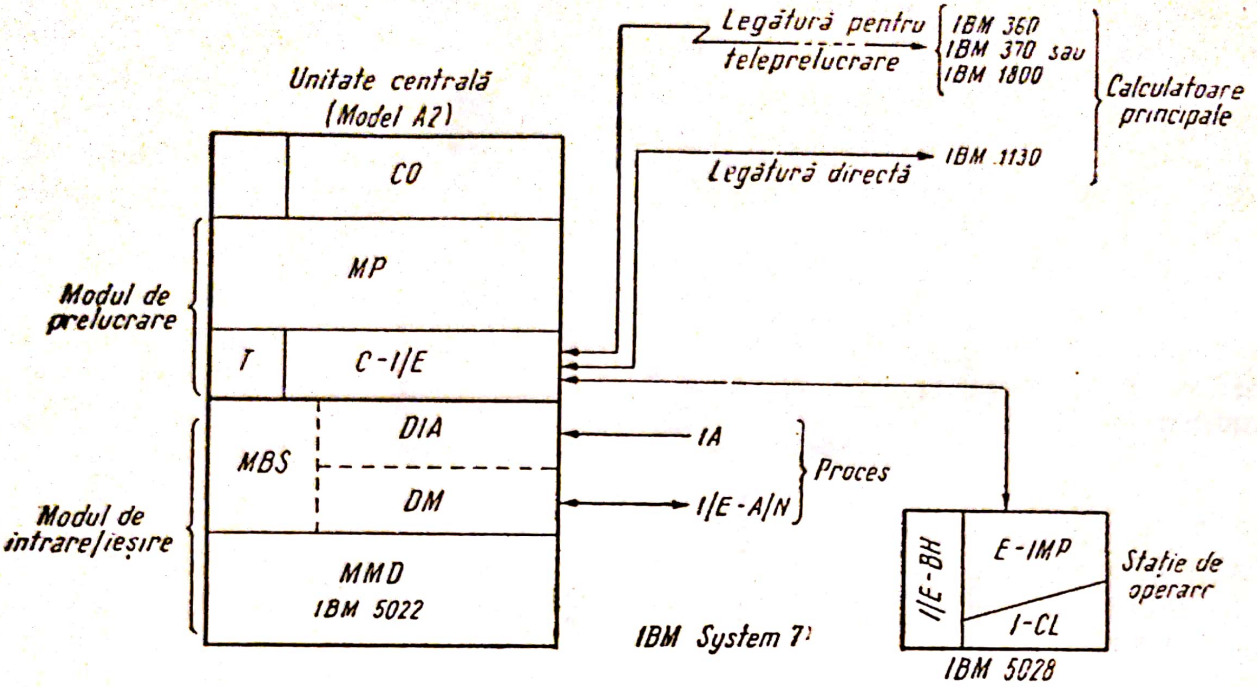


Fig. 11.5.

Prin urmare, componentele de bază ale sistemului IBM S/7 sînt: modulul de prelucrare şi modulele de intrare/ieşire (instalate în unul sau două dulapuri şi formînd unitatea centrală), precum şi o staţie de operare (care este un periferic de intrare/ieşire de tip „flexowriter").

Modulul de prelucrare are un ciclu de timp pentru operaţii de 400 ns şi o memorie principală monolitică formată din 1 la 8 module de cîte 2 048 cuvinte fiecare (deci cu o capacitate de 2 048 pînă la 16 384 cuvinte). Cuvîntul are lungimea de 2 bytes, deci cu 16 bits informaţionali pe cuvînt (la care se mai adaugă cîte 2 bits/cuvînt necesari pentru verificarea parităţii). Sînt prevăzute patru nivele de întreruperi, pentru fiecare existînd 7 registre index, un acumulator şi un registru de adresă instrucţiune.

Modulul de intrare/ieşire este de două categorii de bază: unul ca cel din figura 11.5, care conţine modulul de bază pentru senzori *(MBS)* plus un modul de memorie pe disc *(MMD)*, şi altul care nu are decît modulul de bază pentru senzori (deci fără discul de memorie). Modulul de bază pentru senzori (intrări/ieşiri-proces) are două dispozitive: unul pentru introducerea semnalelor analogice provenite de la tranductoarele şi dispozitivele de supraveghere a procesului tehnologic (*DIA* pe fig. 11.5) şi altul multifuncţional (dispozitivul IBM 5012). Acest dispozitiv multifuncţional (*DM* pe fig. 11.5) conţine patru module: modulul de intrări analogice (IBM 5014) care, printr-un multiplexor, poate fi conectat la 32 puncte de măsurare ale procesului; modulul de ieşiri analogice, prevăzut cu unul sau două canale de transmitere a datelor de reglare pentru unul sau două puncte ale procesului; modulul de intrări numerice, pentru 128 puncte de măsurare ale procesului (constînd în semnale treaptă de tensiune electrică sau contacte) şi un modul de ieşiri numerice pentru 64 puncte de comandă-reglare a procesului prin dispozitive de trei tipuri (cu registre numerice de putere scăzută, cu trenuri de pulsuri de putere medie şi cu contacte de relee lichide cu mercur). Modulul de memorie

Scanned with OKEN Scanner

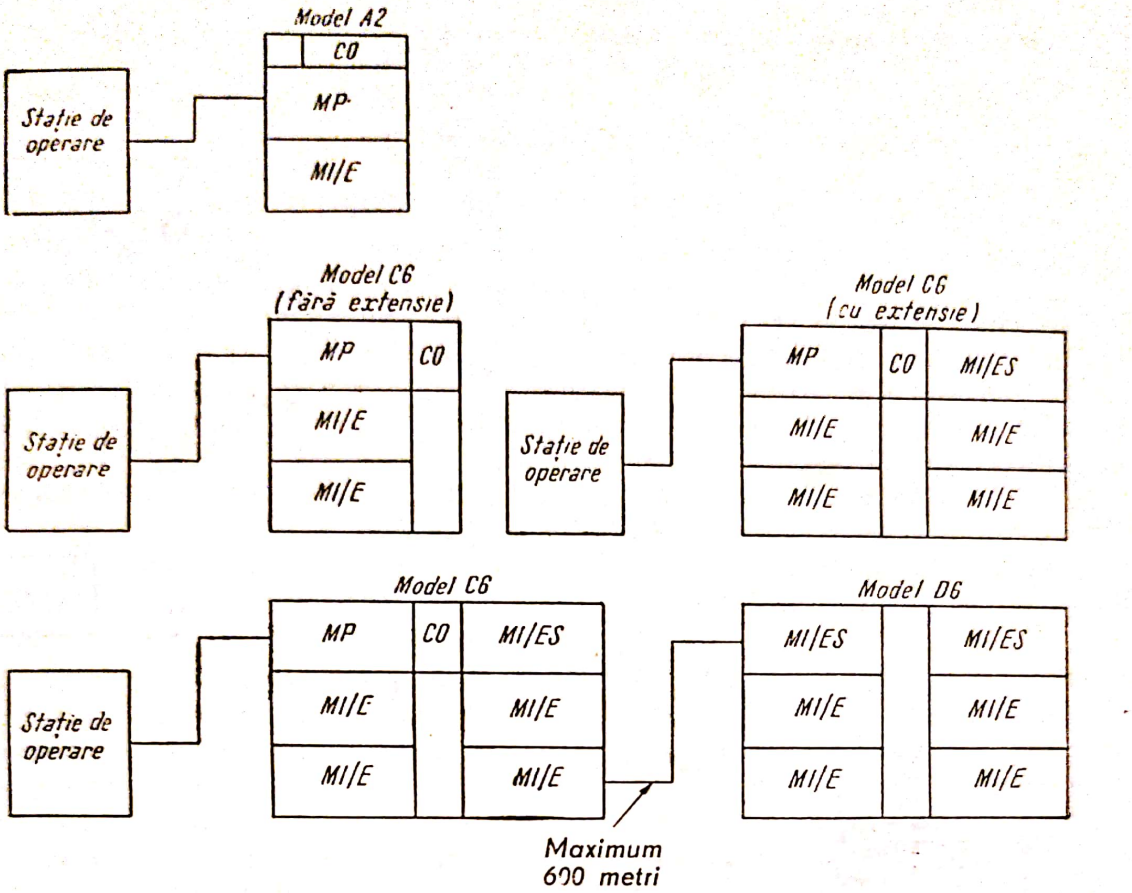


Fig. 11.6.

pe disc (IBM 5022) reprezintă o memorie auxiliară necesară pentru: date, programe, IPL („Inițial Program Load") și „Automatic IPL". Această memorie poate fi folosită și ca o memorie tampon („buffer") pentru comunicațiile dinamice între System/7 și un calculator principal. Se pot realiza patru capacități diferite pentru memoria pe disc: de 1 228 000 cuvinte pe un disc fix; 1 228 000 cuvinte pe două discuri interșanjabile; 2 457 000 cuvinte pe un disc fix și 2 457 000 cuvinte pe două discuri interșanjabile. Timpul de acces este cuprins între 162 ms la 269 ms, cu o viteză de transfer a datelor de 99 500 cuvinte pe secundă.

Datorită construcției sale cu multe module, IBM System/7 are o mare versatilitate, putîndu-se realiza — prin diverse combinații (așa cum este arătat în figura 11.6) — patru configurații standard, care permit o adaptare economicoasă (ca investiție) la numeroase aplicații (ca tipuri de proces și ca anvergură). Notațiile din figura 11.6 au următoarele semnificații: *CO* — consola operatorului, *MP* — modul de prelucrare, *MI/E* — module de intrare/ieșire și *MI/E S* — module de intrare/ieșire de bază pentru senzori (fără memoria pe disc).

IBM System/7, fiind un calculator de proces de mare viteză și cu sistem de operare în timp real, are un larg domeniu de aplicații; el poate fi utilizat:

individual, ca simplu calculator de proces (fig. 11.7, *a*) la supravegherea unui număr mare de mașini sau linii tehnologice de fabricație și la reglarea mai multor procese (continue sau discontinue);

Scanned with OKEN Scanner

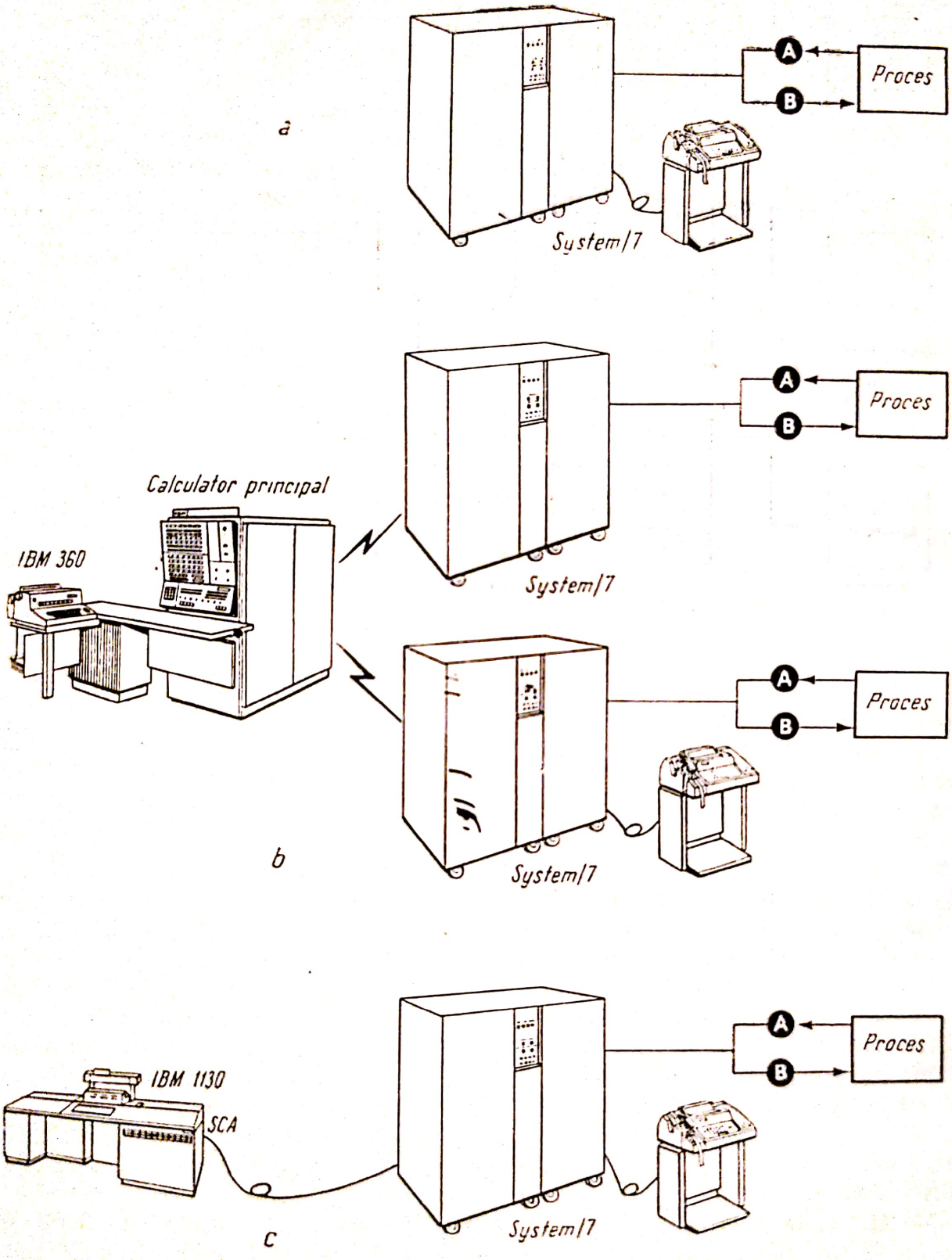


Fig. 11.7.

Scanned with OKEN Scanner

conectat prin linii de telecomunicații la un calculator principal (un IBM 360 sau IBM 370 sau IBM 1800), în acest caz IBM System/7 fiind un terminal de proces (fig. 11.7, *b*);

conectat local la un calculator de gestiune (de exemplu un sistem de calcul IBM 1130), așa ca în figura 11.7, *c*, prin intermediul unui canal de acces la memoria internă (notat cu *SCA* în figura 11.7, *c*).

În figura 11.7, *A* reprezintă intrările în calculator (de la proces) și *B* ieșirile din calculator (către proces).

Ex. 11—3: configurația sistemului de calcul **IBM 1800.** Calculatorul IBM 1800 are o structură suplă, formată din echipamente cu performanțe ridicate, conceput pentru o gamă largă de aplicații în timp real, fiind destinat în special conducerii proceselor industriale și colectării datelor cu mare viteză. Spre deosebire de echipamentele „clasice" (universale) de prelucrare a datelor, calculatorul IBM 1800 a fost conceput cu o arhitectură aparte, astfel încît sistemul să poată înregistra și regla un mare număr de variabile (parametrii de proces), atît pentru instalațiile (grupuri de instalații și uzine) productive, cît și pentru încercările din laboratoarele de cercetare, testare și probe tehnologice.*

În fapt, sistemul IBM 1800 colectează direct semnalele furnizate de „senzorii" procesului, le prelucrează corespunzător unei situații optime impuse de anumite criterii programate și creează în timp real semnalele necesare pentru reglarea procesului (pe care le „emite" elementelor executive din cadrul dispozitivelor de automatizare a procesului), elaborînd totodată și materialele cu înregistrările valorilor parametrilor procesului necesare utilizatorului care urmărește (studiază) procesul. Pe lîngă acestea, calculatorul IBM 1800 poate fi conectat cu un alt calculator (de exemplu cu unul de tip IBM 360 sau IBM 370) în scopul de a mări puterea totală de calcul a sistemului sau de a realiza un sistem integrat pentru gestiunea întreprinderilor în cadrul căruia datele relative la liniile tehnologice de fabricație sînt utilizate pentru ținerea la zi a unor fișiere cu articole referitoare la inventare, stocuri, prețuri, situația producției, aprovizionarea tehnică-materială, relații și contracte cu alte întreprinderi, evidența operativă a consumurilor de materii prime, materiale, energie și combustibili ș.a.m.d.

Calculatorul IBM 1800 permite conectarea simultană a dispozitivelor de intrare/ieșire ale procesului cu funcționare în timp real, cum ar fi: „intrările" analogice, „ieșirile" analogice, analizoarele cu contacte, dispozitivele de comandă/execuție cu contacte, precum și alte sisteme de supraveghere a proceselor (cum ar fi IBM System/7) sau diferite unități „clasice" de intrare/ieșire pentru prelucrarea datelor.

Datorită concepției sale constructive, bazată pe un anumit număr de module funcționale distincte, este posibil să se creeze diferite configurații

---

* Printre aplicațiile curente de reglare a proceselor, pentru care calculatorul IBM 1800 a fost în mod special conceput, enumerăm cîteva: rafinarea petrolului (și în general procesele petrochimice), prelucrarea produselor chimice, laminarea oțelurilor, transportul fluidelor prin conducte, producerea—transportul — distribuția energiei electromagnetice etc. Posibilitățile calculatorului IBM 1800 relativ la colectarea datelor cu mari viteze sînt în mod special utile (eficiente) pentru operațiile preliminare de lansare a rachetelor, pentru controlul producției, pentru cercetările de la reactoarele nucleare, analiza și supravegherea medicală și clinică, cercetările medicale și numeroase alte aplicații.

Scanned with OKEN Scanner

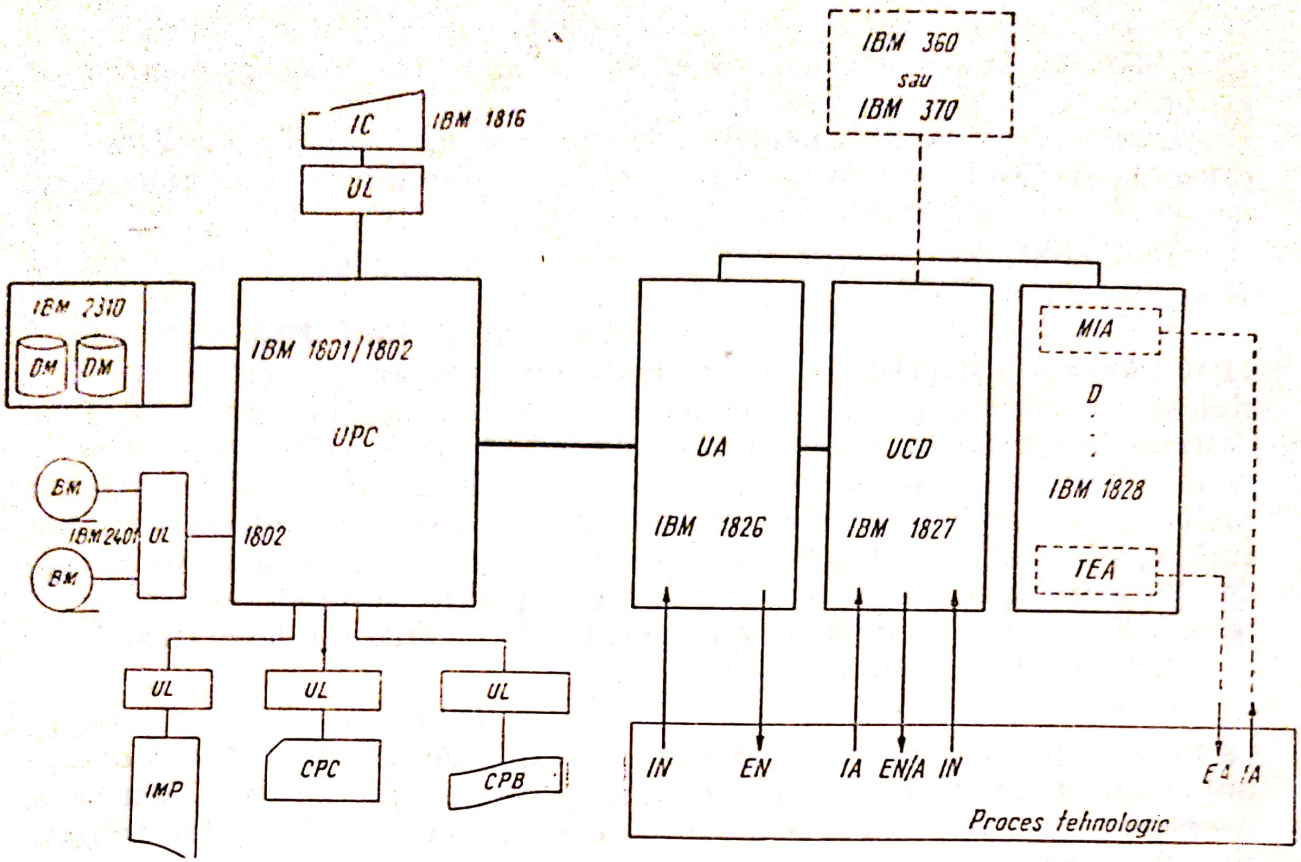


Fig. 11.8.

efective, prin adaptarea configurației standard la specificul oricărei aplicații. Unitățile configurației standard complete (fig. 11.8), pe care un utilizator le poate alege pentru a-și alcătui propria sa configurație de calculator IBM 1800, adaptată unui anume proces, sînt: unitatea de prelucrare și coordonare (*UPC* — în fig. 11.8), care poate fi de tipul IBM 1801 sau/și IBM 1802; mașina de scris (imprimantă) cu claviatură *(IC)* de tip IBM 1816; unitatea de adaptare a datelor *(UA)* de tip IBM 1826, care este o interfață industrială pentru intrările/ieșirile numerice; unitatea de control a datelor *(UCD)* de tip IBM 1827 care este o interfață hibridă standard a firmei IBM (v. *Ex. 11—10*); dulapul metalic IBM 1828 (*D* în fig. 11.8) care este un șasiu cu 19 sertare destinat montării terminalelor multiplexoare pentru intrările analogice *(MIA)* și terminalelor pentru ieșirile analogice *(TEA)*; unitatea de discuri magnetice (*DM*) pentru memoria externă IBM 2310 și periferice standard: imprimante *(IMP)*, cititoare/perforatoare de cartele *(CPC)*, cititoare/perforatoare de bandă de hîrtie *(CPB)* și unități de bandă magnetică (*BM*). În figura 11.8 au mai fost utilizate și următoarele alte notații: *UL* — unități de legătură, *IN* — intrări numerice; *IA* — intrări analogice; *EN* — ieșiri numerice, *EA* — ieșiri analogice și *EN/A* — ieșiri numerice și analogice.

Unitatea de prelucrare și coodonare IBM 1801/1802 poate fi realizată în două modele (unul la care ciclul memoriei principale este de 4μs, celălalt cu un ciclu 2μs). Aceste două modele pot avea patru capacități de memorie diferite: 4 096, 8 192, 16 384 și 32 768 cuvinte, un cuvînt avînd lungimea

Scanned with OKEN Scanner

de 16 bits. Unitatea IBM 1802 are şi unităţile de control necesare pentru conectarea şi funcţionarea echipamentelor periferice de memorie pe bandă magnetică IBM 2401 şi IBM 2402. Fiecare unitate de prelucrare şi coordonare are: trei canale de date (care funcţionează cu viteza de transfer a datelor de pînă la 500 000 cuvinte/secundă în mod continuu, cu un ciclu memorie de 2μs); trei registre index; trei numărătoare pentru intervalele de timp; un set important de instrucţiuni; adresaj indirect al memoriei principale; 12 nivele de întreruperi şi un dispozitiv de protecţie a memoriei.

Unitatea de control a datelor IBM 1827 este o interfaţă standard hibridă destinată asigurării conexiunilor cu dispozitivele de intrare analogică şi numerică de supraveghere a procesului, comparării valorilor de intrare cu anumite date limită (prescrise) şi realizării legăturilor cu dispozitivele de ieşire analogice şi numerice pentru comanda procesului. Această unitate, care va fi prezentată mai în detaliu în cadrul subcapitolului 11.3 (v. *Ex. 11—10*) poate realiza legătura simultană cu pînă la 288 puncte de intrare/ieşire numerice şi asigură interfaţa proces-calculator IBM 1800 prin trei dispozitive de bază: de intrări analogice, de intrări numerice şi de ieşiri numerice şi analogice.

Unitatea de adaptare a datelor IBM 1826 permite realizarea unor legături de intrare/ ieşire numerice în cazul în care necesităţile aplicaţiei depăşesc posibilităţile unităţii IBM 1827. Astfel, una sau mai multe unităţi IBM 1826 sînt necesare dacă numărul punctelor de intrare/ieşire numerice de proces este mai mare decît 288, sau numărul total al adaptoarelor de intrare numerică, al adaptoarelor numărătoarelor de pulsuri şi al adaptoarelor de ieşire numerică depăşeşte cifra trei. Fiecare unitate IBM 1826 poate asigura legătura cu 576 puncte de intrare/ieşire numerice ale procesului şi o combinaţie totală de 6 adaptoare de intrare numerică, de ieşire numerică şi pentru numărătoarele de pulsuri. Fiind posibilă utilizarea a pînă la 6 unităţi IBM 1826 pentru un calculator IBM 1800, rezultă că o configuraţie maximă poate asigura legătura cu 1 024 puncte de intrare numerică, 2 048 puncte de ieşire numerică, 8 adaptoare de intrare numerică şi 32 adaptoare de ieşire numerică.

Caracteristicile calculatoarelor de proces

Din toate cele trei exemple precedente, cititorul a reţinut faptul că deşi calculatoarele de proces sînt — în fond — calculatoare „specializate", totuşi ele rezultă din „adaptarea", prin intermediul unor interfeţe, a aparaturii de măsurare şi reglare/comandă, specifice supravegherii unui proces, la un calculator numeric universal (v. fig. 11.1). Totuşi, chiar în condiţiile unei instrumentaţii de proces corespunzătoare şi a unei interfeţe industriale (proces-calculator) corect alese, nu orice calculator numeric universal automat poate fi utilizat într-un sistem de conducere a proceselor; mai mult, un calculator universal — oricît de puternic ar fi, cu dotări şi resurse deosebite — nu este indicat să fie utilizat la conducerea proceselor decît numai dacă există o justificare economică şi tehnică în favoarea lui, legată de natura şi „anvergura" procesului/instalaţiilor, de complexitatea şi nivelele ierarhizate de conducere urmărite de utilizator,

Scanned with OKEN Scanner

precum și de posibilitățile de folosire a calculatorului și pentru alte aplicații care să-l solicite la nivelul său de dotare.

Din exemplele de pînă aici, ca și din noțiunile introductive prezentate în subcapitolul 10.2, am dedus faptul că, pentru a putea fi utilizat într-un sistem de conducere a proceselor, un calculator numeric universal trebuie să îndeplinească anumite condiții, să aibă anumite caracteristici impuse de acest gen de utilizare (ca, de pildă: multiacces, multiprogramare, lucru în timp real etc.), caracteristici pe care le vom comenta în continuare.

Să mai remarcăm, de la început, că în vederea unei alegeri economice (din punctul de vedere al investiției în echipamentele de calcul și adaptare) pentru diferite genuri de aplicații, cu diverse grade de complexitate și amploare, firmele constructoare produc calculatoare universale cu o arhitectură orientată și către conducerea proceselor, avînd o structură pronunțat modularizată. În acest fel, utilizatorul va putea să realizeze, potrivit specificului aplicației sale, o configurație efectivă minimală ca investiție; apoi, în funcție de dezvoltarea instalațiilor cu procese conduse, precum și pe măsura cîștigării de exprienţă în aplicațiile efectiv realizate, sau de asigurare a condițiilor pentru trecerea și la alte aplicații (gestiune, regăsirea informației, calcule științifice etc.), utilizatorul își va putea dezvolta treptat (prin mici investiții succesive) configurația inițială a calculatorului său de proces. Din același motiv și pentru a permite creșterea eficienței utilizării calculatoarelor de proces în sisteme de conducere ierarhizată, constructorii de calculatoare au prevăzut și produc și echipamentele necesare pentru conectarea unui calculator de proces la un calculator principal, direct sau prin tehnici de telecomunicații. Această caracteristică privind *suplețea*, mai degrabă o cerință impusă echipamentelor de prelucrare automată a datelor, trebuie avută în vedere de orice utilizator și folosită judicios atunci cînd trece la alegerea și implementarea unui calculator de proces.

Cerințele funcționale impuse unui sistem de proces rezultă din faptul că în timp ce procesul reprezintă un sistem fizic de tip analog (cu variabile de stare — parametri fizicochimici — a căror valoare locală variază continuu în timp), calculatorul pus să conducă acest proces este un sistem numeric în care semnalele corespunzătoare parametrilor de stare ai procesului trebuie reprezentate printr-o secvență de numere (deci în formă discretă), obținută prin eșantionarea valorilor semnalelor continue furnizate de instrumentația de redare a stării procesului. De asemenea, supravegherea proceselor fizice ce se desfășoară în timp real trebuie efectuată cu calculatorul numeric, care este în esență o mașină de tip serie, ce prelucrează datele printr-o secvență de operații aritmetice și logice, conform unui program. Ca urmare, cerințele funcționale pentru un sistem de conducere a proceselor sînt determinate de problemele de interconectare (proces „analogic" — calculator numeric) și de control al sistemului însuși, cu aspecte specifice referitoare la: conversia datelor, transferul de date, eșantionarea simultană a datelor, versatilitatea, controlul sistemului etc.

Conversia datelor trebuie să asigure trecerea de la semnalele analogice furnizate de dispozitivele de măsurare a parametrilor procesului la semnale discrete corespunzătoare, apte de a fi preluate și prelucrate de calculato-

Scanned with OKEN Scanner

rul numeric. În acest scop se folosesc echipamente de conversie (analog-numeric și numeric-analog). Convertoarele analog-numerice transformă semnalele furnizate de instrumentația de proces (tensiuni electrice analogice) într-o „intrare" adecvată (numerică) pentru calculatorul de proces; convertoarele numeric-analogice transformă secvențele de numere de la ieșirea calculatorului de proces în semnale analogice (tensiuni sau/și curenți electrici) necesare pentru comanda unor dispozitive de execuție/reglare prevăzute în scopul dirijării procesului. De obicei, lungimea cuvintelor convertite și transmise trebuie să fie de cel puțin 15 bits plus bitul de semn, iar echipamentele de conversie trebuie să aibă o clasă de precizie globală de 0,1.

Transferul de date între proces și calculator trebuie să fie rapid și eficient (în sensul realizării simultaneității unor operații de transfer). Se știe că ciclul de prelucrare a datelor într-un sistem de proces constă din patru operații de bază: „ridicarea" datelor de proces (sau „folosirea" lor pentru dirijarea procesului); transmiterea de date analog-numerică; calcule numerice și/sau logice în unitatea de prelucrare; transmiterea de date numeric-analoagă. Pentru mărirea eficienței transmisiei și prelucrării, în multe cazuri fazele a două sau mai multe operații de bază se pot realiza logic simultan. Pentru aceasta, apare necesitatea ca sistemul de calcul să accepte suprapunerea operațiilor de intrare-ieșire cu operațiile de calcul și, uneori, chiar cu operațiile de conversie și multiplexare.

Eșantionarea simultană a datelor. Sînt destule cazurile în care supravegherea unui proces impune ca datele analogice să fie eșantionate simultan în toate punctele de ieșire din proces. În aceste cazuri, o soluție posibilă ar fi explorarea punctelor de ieșire din proces cu o viteză mai mare decît viteza de eșantionare; astfel se reduc, la un nivel acceptabil, efectul tranzitoriu și eventualele decalări. O altă soluție (și cea mai indicată) constă în folosirea unor memorii tampon („buffer"-e) pe ieșirile instrumentației de proces; aceste „buffer"-e extrag valorile instantanee și le păstrează pînă la terminarea conversiei analog-numerice. Tot așa, în multe cazuri, este necesar ca toate comenzile pentru proces elaborate de calculator să fie prezentate simultan la intrarea în proces; sînt posibile două soluții: fie „buffer"-e numerice la intrarea convertoarelor numeric-analogice, fie „buffer"-e analogice la ieșirea din convertoare (prima soluție este de preferat deoarece o memorie tampon analogică este susceptibilă de derivă — „drift").

Versatilitatea sistemului. Pentru a putea opera eficient într-un domeniu larg de probleme, sistemul calculator-interfață de proces trebuie să dispună de „abilitatea" de a varia vitezele de eșantionare, de a modifica punctele de intrare/ieșire și de a explora diversele puncte cu viteze de eșantionare diferite. Toate acestea conduc la necesitatea posibilității de adresare selectivă pentru intrările și ieșirile de proces ale calculatorului.

Controlul sistemului. În fiecare ciclu (măsurare/comenzi, intrare/ieșire, conversii și calcule) este necesar să se controleze inițierea operațiilor de intrare/ieșire, conversie și calcule; un control corespunzător trebuie să se exercite și asupra instrumentației de proces (aparatură de măsurare, reglare și comandă/execuție). Sarcina aceasta, de a emite comenzile pentru controlul sistemului, revine calculatorului; cu toate acestea, este necesar ca și „evenimentele" apărute în cadrul procesului să poată întrerupe cal-

Scanned with OKEN Scanner

culatorul (întreruperi externe) şi astfel să determine o modificare a secvenţei calculelor din unitatea centrală de prelucrare. Pentru aceasta, sincronizarea operaţiilor se realizează, cel mai comod, cu un „timer" (un orologiu generator de tact şi numărător de intervale de timp), al cărui ciclu de bază este reglabil în funcţie de ciclul de lucru al sistemului. Semnalele generate de „timer" pot iniţia operaţiile de măsurare/reglare din proces sau pot fi utilizate ca semnale de întrerupere pentru unitatea centrală de prelucrare.

Condiţiile funcţionale impuse calculatoarelor de proces sînt, în general, uşor realizabile cu echipamentele de prelucrare automată a datelor din generaţiile a III-a şi a IV-a; aceste condiţii sînt reunite în mod eficient în special de echipamentele de calcul din familia aşa-numitelor minicalculatoare şi microcalculatoare (asupra cărora vom reveni ceva mai încolo). Condiţiile impuse calculatoarelor numerice de utilizarea lor la conducerea proceselor tehnologice sînt, pe scurt, următoarele:

— viteză de prelucrare mare (cu durata unei operaţii „registru-registru" şi „registru-memorie" de ordinul fracţiunilor de microsecundă ca, de exemplu, 0,4 la 0,8 μs);

— memorie internă de mare capacitate pentru calculatoarele principale, din reţelele pentru conducerea ierarhizată;

— memorie externă cu acces direct (cu discuri magnetice);

— protecţia memoriei;

— canale de intrare/ieşire independente;

— sistem evoluat de întreruperi, cu mai multe nivele de întreruperi ierarhizate, care să permită întregului sistem de proces lucrul în timp real;

— set complet de dispozitive periferice standard (inclusiv perifericele de afişaj — de tip „display" şi de trasare a graficelor — de tip „plotter", utile în special pentru controlul proceselor de laborator şi de cercetare);

— unităţi de control specializate pentru dirijarea teletransmiterii datelor;

— unităţi de control specializate pentru interconectarea sistemelor (de tip DIRECT CONTROL).

Dintre resursele de mai sus, cele mai importante pentru crearea unui sistem de conducere automată a proceselor sînt: canalele de intrare/ieşire, „controlul direct" şi întreruperile externe.

Canalele de intrare/ieşire. Acestea asigură o cale de transmitere a datelor şi operaţiilor de control pentru comunicarea între dispozitivele periferice (standard şi de proces) şi unitatea centrală de prelucrare. Importanţa sistemului de intrare/ieşire pentru conducerea unui proces este atît de mare încît asupra acestui sistem vom reveni, în detaliu, în cadrul subcapitolului 11.3; acum ne vom limita la prezentarea succintă a funcţiunilor de bază ale canalelor de intrare/ieşire.

În general, aceste canale operează asincron şi independent faţă de unitatea centrală de prelucrare (pe care, pentru simplitate, o vom denumi, pe scurt, procesor), eliberînd-o de sarcina de a comunica direct cu dispozitivele de intrare/ieşire şi, prin aceasta, făcînd, posibilă suprapunerea (simultaneitatea) prelucrării datelor cu operaţiile de intrare/ieşire

Scanned with OKEN Scanner

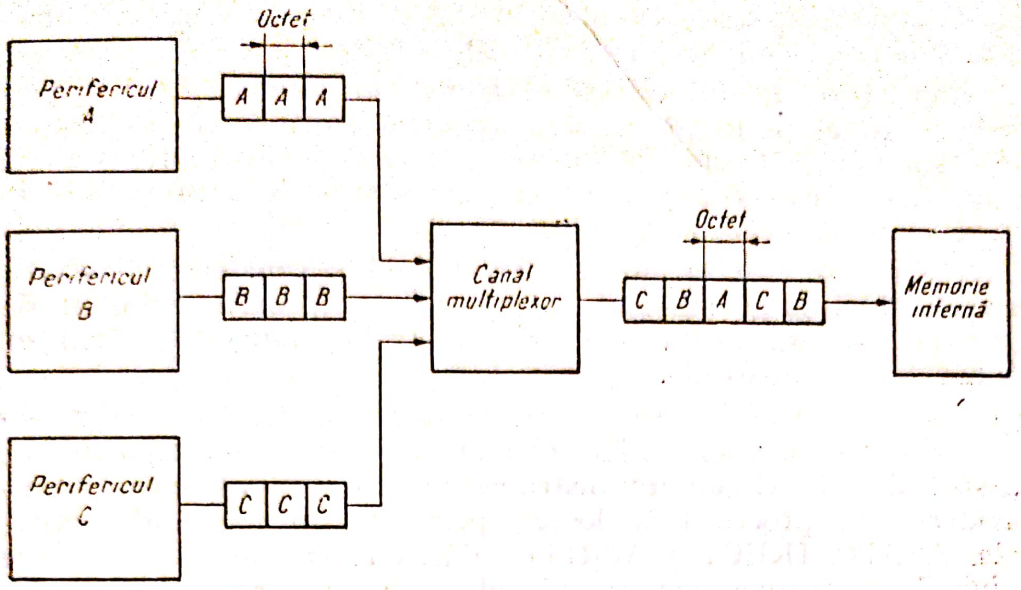


Fig. 11.9.

După cum se știe, sistemele de prelucrare automată a datelor din generația a III-a și a IV-a dispun de două tipuri de canale: multiplexor și selector. *Canalul multiplexor* poate fi în legătură simultană cu mai multe periferice lente lucrînd cu acestea în regimul de divizare a timpului prin intercalare (adică așa-numitul „data interleave mode"), așa cum este indicat în schema din figura 11.9. *Canalul selector* lucrează cu viteze de transfer a datelor mult mai mare (de exemplu: 256 000, ..., 1 300 000 caractere/secundă), el stabilind numai legături individuale „directe" cu dispozitivele periferice rapide, pentru transmiterea unor blocuri de date în forma redată schematic în figura 11.10 (adică așa-numitul „burst mode"). Pe toată durata transmiterii blocului de date, facilitățile de transfer ale canalului selector sînt „monopolizate" de perifericul conectat, celelalte

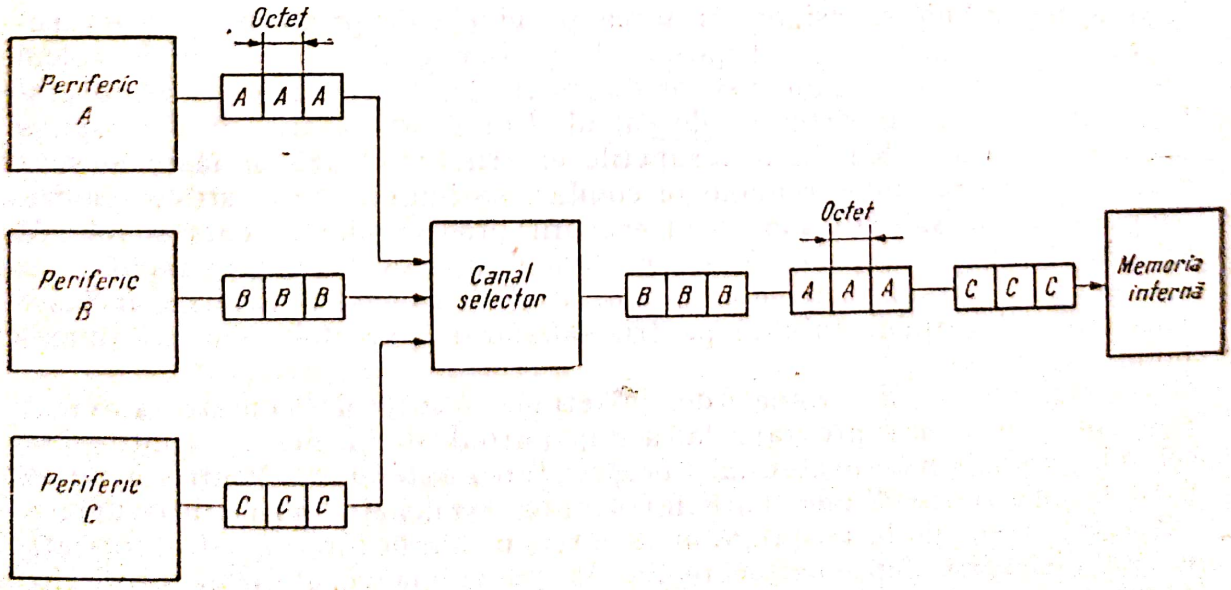


Fig. 11.10.

periferice ataşate aceluiaşi canal neputînd transmite date pînă cînd nu se termină blocul precedent (v. fig. 11.10).

Indiferent de modul de lucru, transferul de date între dispozitivele periferice şi canale se face în modul „caracter după caracter" („octet după octet"). Canalele „răspund" la un set standard de instrucţiuni şi comenzi de intrare/ieşire, care compun partea de intrare/ieşire a programului calculatorului numeric.

Aşa cum vom arăta în subcapitolul 11.3, canalele de intrare/ieşire oferă posibilităţi largi (absolut indispensabile) pentru schimbul de date între procesor şi dispozitivele periferice de proces, bineînţeles prin intermediul interfeţelor industriale.

Controlul direct şi întreruperile externe. Dispozitivele de control direct permit stabilirea unei legături directe între procesor şi dispozitivele periferice speciale (de exemplu: instrumentaţia de supraveghere, reglare şi comandă a unui proces tehnologic), prin utilizarea a două instrucţiuni speciale, READ DIRECT şi WRITE DIRECT, precum şi prin intermediul unor linii de întreruperi externe (de obicei în număr de şase).

Instrucţiunile de citire (intrare) şi scriere (ieşire) directă permit transferul direct al unui caracter (octet) între dispozitivele periferice speciale (aparatura de proces) şi memoria internă, fără intermediul canalelor. Liniile de întrerupere permit stabilirea condiţiilor de întrerupere externă a procesorului, sub acţiunea unor semnale generate de dispozitivele periferice speciale, de proces (v. fig. 11.2) şi/sau de alt procesor (în cazul unei scheme cu mai multe calculatoare interconectate, aşa cum este exemplul din figura 11.7, *b*).

În mod uzual, transferul de date este realizat prin intermediul canalelor (v. fig. 11.9 şi fig. 11.10), iar controlul direct este utilizat pentru schimbul informaţiei de control şi sincronizare între procesor şi perifericele speciale (ale procesului industrial).

Unitatea centrală de prelucrare este prevăzută cu un sistem evoluat de întreruperi care să permită sistemului lucrul în timp real. *Sistemul de întreruperi* trebuie să asigure tratarea pe nivele de prioritate a întreruperilor externe generate de dispozitivele periferice ale sistemului de calcul (ataşate procesului condus) şi de dispozitivele de comunicaţii ale operatorilor tehnologi cu sistemul de calcul. Dat fiind că timpul de răspuns al sistemului de calcul la întreruperile externe afectează în mare măsură desfăşurarea procesului tehnologic condus, se impun limite stricte pentru timpul necesar salvării sau/şi întreruperii programului în curs şi reluării sale după satisfacerea cererii externe (maximum $15 \div 25\mu s$); timpul necesar recunoaşterii dispozitivului de proces şi a solicitării externe, inclusiv „demararea" rutinei stabilite pentru satisfacerea acestei solicitări (maximum $3 \div 5$ μs).

Generarea unui semnal de întrerupere externă comută automat procesorul de la un program la altul, putînd — în acelaşi timp — să schimbe şi unele mesaje esenţiale despre starea sistemului. Pentru a nu se permite „distrugerea" unor date importante, atunci cînd prin întreruperea externă se trece de la un program la altul, memoria internă este protejată de un dispozitiv (opţional) care inhibă accesul la zonele ocupate de alte programe.

Scanned with OKEN Scanner

Insistînd puţin asupra întreruperilor, vom reaminti că există trei cauze ce pot determina întreruperea programelor:

— *întreruperi interne*, care provin de la dispozitivele de control interne ale unităţii centrale de prelucrare (ca, de exemplu: controlul validităţii, controlul parităţii, tentativele de scriere într-o zonă protejată a memoriei interne etc.);

— *întreruperi de intrare/ieşire*, datorate dispozitivelor periferice standard, care sînt produse de necesităţile de citire/scriere a informaţiei;

— *întreruperile externe*, care — aşa cum am mai arătat — provin de la dispozitivele de control ale procesului tehnologic.

Ceea ce este mai important, pentru o conducere corespunzătoare a procesului industrial, este clasarea tuturor tipurilor de dispozitive de proces pe diverse nivele de prioritate ierarhizate în funcţie de importanţa lor într-o anumită instalaţie, fapt care va determina „servirea" lor în ordinea prestabilită. Abaterile care au loc în proces vor fi imediat comunicate calculatorului, care va întrerupe programul în curs de rulare şi va trece la executarea altor programe în stare să rezolve noua situaţie survenită. Dacă din partea procesului sosesc simultan mai multe solicitări, atunci acestea vor fi „tratate" în ordinea de priorităţi după care au fost departajate între ele diversele elemente structurale ale instalaţiei (după importanţa lor funcţională).

În sistemele de conducere a proceselor, controlul direct şi întreruperile externe (ordonate după priorităţi) oferă mijloacele de sincronizare a sistemului, de modificare a secvenţei programului în funcţie de evenimentele apărute în proces (instalaţie) şi de comunicare directă între procesor şi operatorul tehnolog.

Ex. 11—4. Vom ilustra cele de mai sus prin modul cum se realizează controlul direct la calculatorul de proces FELIX C—32P (v. *Ex. 11—1*). În acest caz, schimbul de informaţie între proces şi unitatea centrală de prelucrare se face sub controlul programului cu ajutorul instrucţiunilor de citire directă (simbolizată prin RD) şi de scriere directă (WD).

Fiecare instrucţiune RD şi WD permite transferul unui cuvînt cu lungimea de 32 bits; transferul se face între registrul general *R* (definit de instrucţiune) şi dispozitivul de proces selectat cu ajutorul cifrelor binare *Y 16* şi *Y 19* ale adresei *Y* calculate (adresă ce trebuie să fie cuprinsă între 13 şi 15, deoarece în caz contrar unitatea centrală interpretează această instrucţiune ca pe un control intern). Instrucţiunea RD permite citirea unui cuvînt de 32 bits trimis de către dispozitivul din proces ce are adresa *Y* (calculată prin instrucţiune). Instrucţiunea WD permite transmiterea spre dispozitivul din proces cu adresa *Y* (calculată prin instrucţiune) a cuvîntului de 32 bits conţinut în registrul general *R* desemnat de instrucţiune.

Diagrama de timp a instrucţiunii de citire directă RD este prezentată în figura 11.11, iar derularea instrucţiunii de scriere directă WD în figura 11.12. În aceste figuri au fost folosite notaţiile următoare:: *H1M* — semnalul de tact (cu frecvenţa de 1MHz) trimis de unitatea centrală (fazele de derulare a instrucţiunilor au fost notate prin intervalele de tact *H01* la *H19*); *TR* — timpul de răspuns, care depinde de periferic conectat (în figura 11.11, $TR > 250$ ns ceea ce provoacă patru faze *H18*;

Scanned with OKEN Scanner

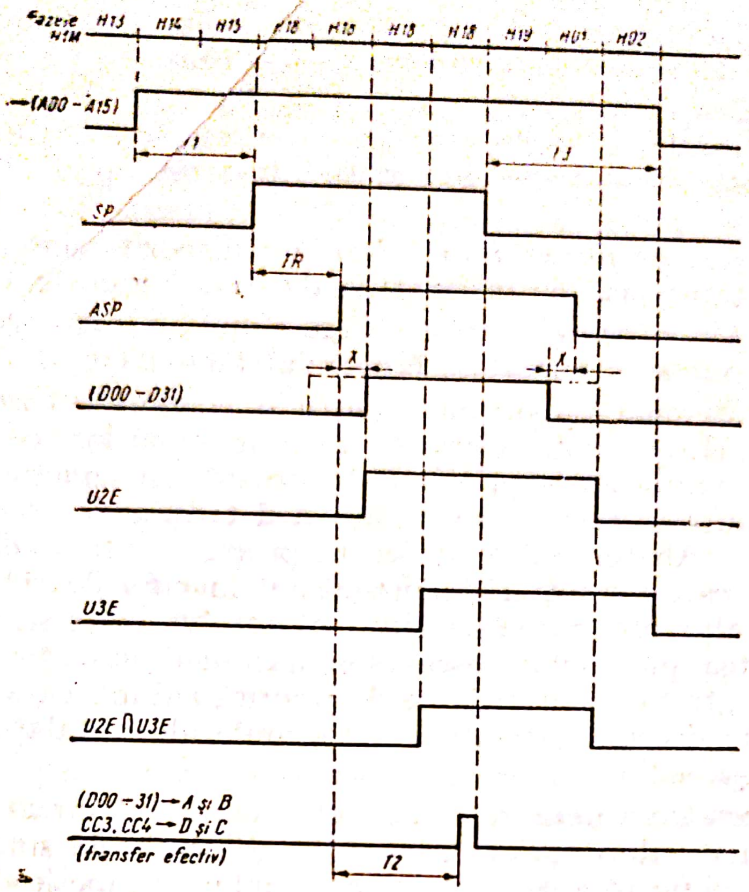


Fig. 11.11.

în figura 11.12, $TR < 250$ ns ceea ce provoacă trei faze $H18$); $T1$ — — intervalul de timp de 500 ns în figura 11.11 și 350 ns în figura 11.12; $T2 \geqslant 500$ ns; $T3 = 750$ ns; $X =$ interval de timp ce depinde de lungimea legăturii fizice. Celelalte notații se referă la semnalele interfeței legăturii programate cu dispozitivele specializate de proces (v. fig. 11.3) și au următoarele semnificații:: *D00* la *D31* — semnale bidirecționale de informație (*D00* corespunde cifrei binare cu ponderea cea mai mare); *A00* la *A15* — semnale de adresă; *SP* — semnal martor trimis de unitatea centrală, care permite luarea în considerare de către dispozitivul adresat a adresei și a datelor ce trebuie transmise pe firele de informație; *ASP* — semnal acceptor martor, trimis de dispozitivul din proces adresat, care indică preluarea adresei prezente pe firele de adresă (în cazul unei ieșiri de informație către dispozitiv, semnalul *ASP* preia datele prezente pe firele de informație, iar în cazul unei intrări de informație dinspre dispozitiv spre unitatea centrală de prelucrare, semnalul *ASP* plasează datele pe firele de informație); *CC3* — semnal ce poziționează indicatorul *D* din registrul „stare program" (este un semnal transmis de dispozitiv cu semnificația: „dispozitiv nerecunoscut"); *CC4* — semnal care poziționează indicatorul *C* din registrul „stare program" (este un semnal transmis tot de către dispozitiv cu semnificația: „dispozitiv ocupat"). Semnalele *U2E*, *U3E* și conjuncția lor $U2E \cdot U3E$ sînt semnale de control ale legăturii directe.

Așa cum rezultă din figurile 11.11 și 11.12, legătura se face prin emiterea adresei *Y* (a dispozitivului din proces) de către unitatea centrală.

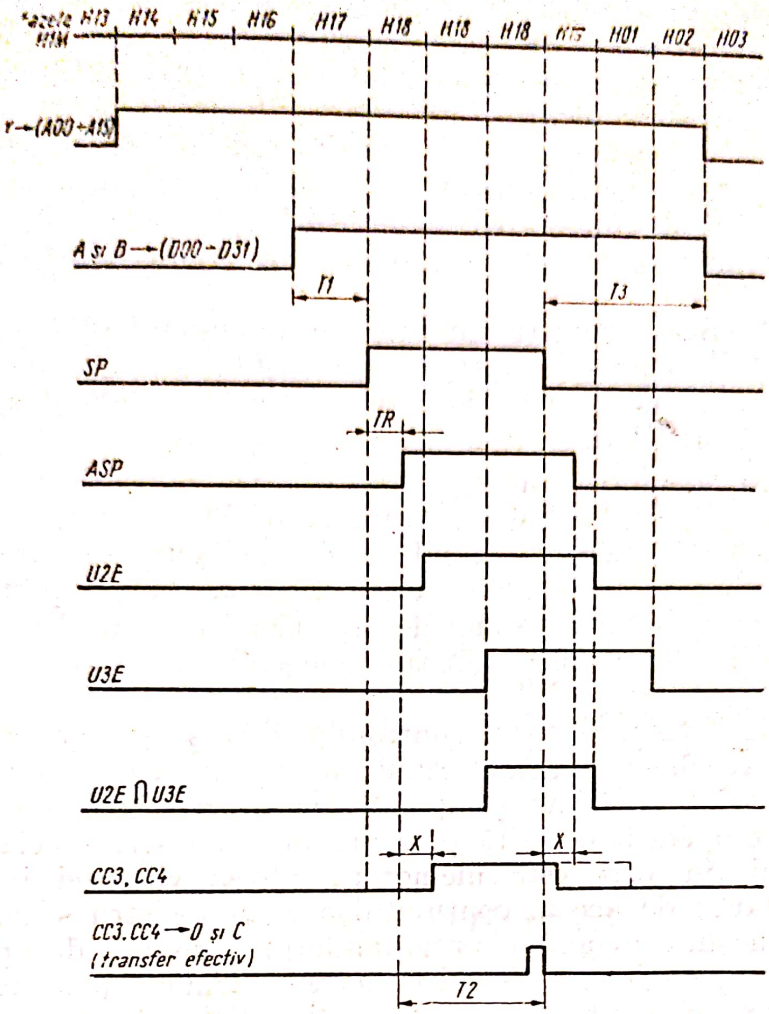


Fig. 11.12.

Tot unitatea centrală de prelucrare emite şi un semnal *RWD* care precizează sensul schimbului de informaţie şi anume: *RWD* = „0" reprezintă „intrare date în unitatea centrală" şi *RWD* = „1" reprezintă „ieşire informaţie din unitatea centrală". Unitatea centrală de prelucrare mai transmite şi un semnal de iniţializare (*RZES*).

Apoi, pentru instrucţiunea WD de ieşire directă (v. fig. 11.12) sînt emise datele *A* şi *B*. După un timp *T1*, de la emiterea adresei *Y* în cazul instrucţiunii RD de intrare (v. fig. 11.11) sau de la începerea transmiterii datelor în cazul instrucţiunii WD de ieşire (v. fig. 11.12), este emis semnalul *SP*. După un timp de răspuns *TR* (a cărui durată, aşa cum am mai arătat, depinde de dispozitivul periferic de proces conectat), este „recepţionat" semnalului martor *ASP* de acceptare; apoi, după un timp $T2 \geqslant 500$ ns, unitatea centrală preia/depune datele, poziţionează indicatorii de condiţie *D* („nerecunoscut") şi *C* („ocupat"), şi suprimă (după realizarea transferului efectiv) semnalul martor *SP*. Pentru instrucţiunea WD (fig. 11.12), adresa *Y* şi datele *A* şi *B* rămîn prezente pe firele de adresă şi de informaţie un timp $T3 = 750$ ns după „dispariţia" semnalului martor *SP*.

Unitatea centrală de prelucrare. Procesorul conține : o memorie (memoria internă foarte rapidă, cel mai adesea cu ferite sau cu semiconductoare de tipul MOS) ; registrele necesare pentru operațiile aritmetice și pentru adresare ; blocul pentru operațiile aritmetice și logice ; blocul de comandă al sistemului de echipamente și ale intrărilor/ieșirilor de proces (analogice și numerice) ; registrele de comandă ale canalelor de intrare/ieșire ; dispozitivele sistemului de întreruperi (cu 6÷24 nivele de întrerupere) ; trei contoare de timp („timer"-e) ; dispozitive de supraveghere a funcționării ș.a.

Procesorul poate prelucra cuvinte cu lungime fixă (de obicei de 16 bits, plus 2 bits de control). Operațiile de intrare/ieșire prin canale se pot efectua cu viteza de 25 000÷500 000 cuvinte/secundă (cu un ciclu de extragere a informației din memorie de 0,2 μs÷0,8 μs).

Capacitatea memoriei interne poate varia, după tipul unității centrale de prelucrare, de la 4 096 cuvinte la 65 536 cuvinte (sauplimentar, se poate prevedea o memorie auxiliară de 256 cuvinte care să permită rularea unor programe de depanare a calculatorului fără a perturba rularea programelor de exploatare). Modul de organizare a memoriei permite accesul la date fie direct, fie prin adresare relativă (cu ajutorul unor registre de indexare).

Programele corespunzătoare controlului diverselor elemente de proces aflate în supravegherea calculatorului se prezintă ca o secvență de instrucțiuni, care sînt analizate și apoi executate de către blocul aritmetic și logic. Aceste operații necesită prezența unor registre speciale : registrul de instrucțiuni (în care este memorată adresa viitoarei instrucțiuni ce trebuie executată ; de aceea, conținutul acestui registru se modifică permanent, pe măsura evoluției programului) ; registrul de operație (care primește codul operației) ; registrul de adrese memorie (care indică adresa locațiilor ce conțin datele de prelucrat) și registrele de operații (în care sînt „aduse" datele din memoria internă în vederea prelucrării lor).

Blocul de comandă la procesorului este indicat să fie de tip macroprogramat (și, opțional, microprogramabil conform cerințelor utilizatorilor) ; în acest fel se asigură următoarele exigențe specifice aplicațiilor de conducere a proceselor : flexibilitatea setului de instrucțiuni (ceea ce permite optimizarea acestuia în funcție de tipul aplicației) ; posibilitatea de a adapta un sistem de calcul dat la pachete de programe de aplicații specifice existente sau disponibile la alte tipuri de calculatoare (prin „emularea" arhitecturii acestora) ; compatibilitatea mai multor tipuri de echipamente în cadrul unor sisteme distribuite sau ierarhizate ; sporirea substanțială a vitezei de execuție a unor rutine frecvent solicitate în cadrul unei aplicații concrete, precum și a unor operații elementare (înmulțiri, împărțiri, calcule în virgulă mobilă, calcule în sistem zecimal ș.a.).

Minicalculatoarele. Echipamentele de calcul din familia așa-numită a minicalculatoarelor întrunesc, într-un mod economic și eficient, toate caracteristicile de bază impuse unor calculatoare de proces. Pe scurt, acestea sînt următoarele : efectuarea „în paralel" deci cu viteze foarte mari a tuturor operațiilor aritmetice și logice de bază (cu un set de 64÷120 sau chiar 200 instrucțiuni) ; folosirea blocurilor de comandă microprogramabile chiar de către utilizator ; realizarea operațiilor de transfer cu o gamă largă de tipuri de echipamente periferice standard și speciale (de proces și de tele-

Scanned with OKEN Scanner

prelucrare); operarea cu unităţi informaţionale de prelucrare (cuvinte) cu lungimea de 8, 12, 16 sau 32 bits; structură internă specifică, relativă simplă, realizată cu un număr mic de module funcţionale (de bază şi opţionale) ce se interconectează prin una sau două magistrale universale (căi comune de date şi instrucţiuni); structură flexibilă şi diversificată a „periferiei" de intrare/ieşire. Un exemplu de minicalculator este IBM System/7 (v. *Ex. 11—2*) sau PDP Massachusetts (de tipul celor utilizate pentru controlul proceselor de laborator la Centrul Naţional de Fizică — Măgurele).

Minicalculatoarele prezintă avantaje importante ca: mare flexibilitate (prin utilizarea, după necesităţi, a unor module funcţionale diverse ca tip şi număr); simplitate constructivă; tipizare pronunţată a subansamblurilor lor funcţionale şi constructive; cost redus de fabricaţie; operare simplă etc.

Microcalculatoarele. Acestea sînt echipamente care posedă toate caracteristicile calculatoarelor electronice, însă — datorită condiţiilor tehnologice de realizare a lor (cu un minim de module fizice şi funcţionale, circuite integrate pe scară largă de tip LSI ş.a.), prezintă anumite limitări în ceea ce priveşte arhitectura, posibilităţile de programare, de extindere şi alte performanţe, în comparaţie cu echipamentele din clasa minicalculatoarelor. Totuşi, microcalculatoarele sînt foarte economice, deoarece pot realiza conducerea unor procese tehnologice sau de laborator din instalaţii restrînse cu investiţii minime şi rezultate egale cu ale echipamentelor obişnuite.

11.2. PRELEVAREA DATELOR

În cadrul acestui subcapitol ne vom referi la modalităţile specifice calculatoarelor de proces de *„extragere" (prelevare) a datelor* referitoare la parametrii de stare ai procesului condus, precum şi de *„introducere" a comenzilor* „elaborate" de calculator în vederea modificării parametrilor prin care se realizează conducerea procesului. Nu ne propunem să tratăm în mod deosebit aparatura („instrumentaţia") de proces, prin care se extrag datele sau se introduc comenzile; această aparatură, dacă face parte din structura dispozitivelor de automatizare convenţională, a fost studiată la cursurile de automatică [10], iar dacă este de construcţie specială reprezintă cazuri particulare — specifice unor aplicaţii concrete — care, deci, nu ar putea fi prezentată decît cu titlu de exemplu, ceea ce vom şi face la finele acestui capitol (v. *Ex. 11—11* şi *Ex. 11—12*). De altfel, noţiunile de bază referitoare la instrumentaţia de proces — considerată ca dispozitive periferice speciale de intrare/ieşire ale calculatorului de proces — au fost prezentate în subcapitolul 10.2 (v. pag. 37, ..., 39).

După cum se ştie, operaţiile de conducere a proceselor cu ajutorul sistemelor de calcul electronice automate, operaţii care se efectuează prin intermediul *semnalelor** de natură electrică sau magnetică (ce exprimă

---

* În cadrul acestui subcapitol, vom da noţiunii de semnal accepţiunea mai largă din teoria transmisiunii informaţiei, adică aceea de „manifestare fizică (de exemplu: undă electromagnetică, curent electric, tensiune electrică la borne etc.) capabilă să se propage printr-un mediu dat".

Scanned with OKEN Scanner

codificat datele, adică valorile parametrilor specifici procesului), sînt de patru tipuri distincte:

— extragerea datelor din proces, transmiterea şi introducerea lor în calculator;

— stocarea unora dintre date (cu crearea, menţinerea şi exploatarea unor fişiere);

— prelucrarea datelor (cu determinarea unor valori necesare aprecierii stării procesului sau/şi elaborarea unor comenzi de conducere în timp real a procesului);

— extragerea din calculator a datelor/comenzilor rezultate din faza de prelucrare şi „valorificarea" lor în sensul conducerii procesului.

Dintre aceste operaţii, în cadrul prezentului subcapitol, ne vom referi numai la cele cu privire la extragerea datelor din proces, transmiterea lor şi introducerea în proces a comenzilor, avînd în vedere numai aspectul tehnic (al semnalului, considerat ca suport de date, şi al aparatului de proces, care generează sau recepţionează semnalul). În scopul simplificării terminologiei şi generalizării, vom folosi următoarele două denumiri: *senzori* pentru aparatele din proces care generează semnale ce exprimă valoarea parametrilor de stare ai procesului (ca: traductoare, aparate de măsurat etc.) şi *executori* pentru aparatele din proces care valorifică semnalele de comandă furnizate de calculator (elemente de execuţie, organe de reglare, aparate de semnalizare optice sau acustice etc.). Conform acestor denumiri, senzorii reprezintă instrumentaţia de extragere (prelevare) a datelor referitoare la proces, iar executorii-instrumentaţia de introducere (valorificare) în proces a datelor furnizate („restituite") de calculator.

Senzori — extragerea datelor

Prin activitatea de identificare şi modelare a proceselor (v. subcapitolele 10.3 şi 10.4), rezultă că starea unui proces dat poate fi determinată prin intermediul unui anumit număr de parametri fizico-chimici sau de „condiţii" ale procesului, a căror valoare instantanee şi locală (în diverse puncte ale procesului, stabilite iniţial) trebuie măsurată. Măsurarea se face cu ajutorul senzorilor, cîte unul pentru fiecare parametru şi fiecare punct din proces (topologic distinct).

La modul general, un senzor este format din: un *element sensibil* [10], care realizează fizic acţiunea de prelevare a parametrilor; un *generator de semnale*, care produce un semnal (electric, magnetic sau fluidic) purtător al datelor ce exprimă valoarea parametrului măsurat; un *convertor* de natură a semnalului (care transformă semnalul produs de generator într-un semnal electric, curent sau tensiune); un *amplificator*, de cele mai multe ori de putere. Unele din aceste părţi componente (ca, de exemplu, convertorul sau/şi amplificatorul, uneori chiar şi generatorul de semnale —, dacă elementul sensibil este un traductor de tip generator ca, de pildă, un cuplu termoelectric) pot lipsi din structura unui anumit tip de senzor.

Semnalul produs de senzor, care trebuie să fie neapărat un semnal electric, este transmis din locul de prelevare (situat „în proces") la punctul de conexiuni („cutie cu borne" — cleme de conexiuni), printr-o linie de transmisie electrică (cabluri multifilare), aşa cum este arătat schematic

Scanned with OKEN Scanner

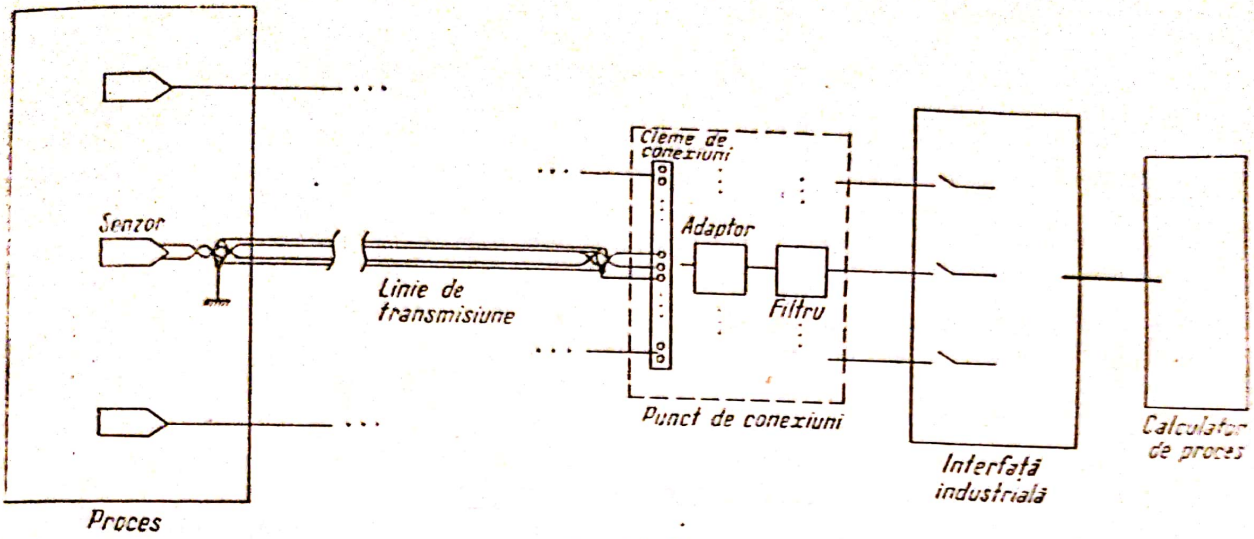


Fig. 11.13.

în figura 11.13. În acest loc (un „dulap" cu „ieşirile" din proces), se face şi „pregătirea" semnalului pentru transmiterea lui la interfaţa industrială proces-calculator, prin adaptoare şi filtre. *Adaptarea* constă fie în amplificarea (uneori atenuarea) nivelului semnalului, fie într-o adaptare la impedanţa de intrare a interfeţei proces-calculator. *Filtrarea* constă în atenuarea componentelor de frecvenţe nedorite, care provin din zgomotele şi distorsiunile suferite de semnal.

În interfaţa proces-calculator, semnalele sînt supuse unor transformări şi transmiteri ce vor fi prezentate, pe larg, în subcapitolul următor — 11.3. Oricum, „ajunse" în interfaţa industrială, semnalele provenite de la diverşi senzori din proces sînt „selectate" prin comutare aleatoare sau secvenţială (sub comanda calculatorului de proces) şi „dirijate" către diverse convertoare sau alte tipuri de adaptoare şi dispozitive de codare.

Semnalele referitoare la starea procesului produse de senzori sînt de tip „on—line", deoarece senzorii sînt, de fapt, periferice speciale (de proces) „aliniate" la unitatea centrală de prelucrare. În afara acestor semnale „on-line", există şi semnale aşa-zise „off-line" introduse manual de către operatori (prin: consola calculatorului, consola operatorului tehnolog, unităţi de citire standard etc.), care se referă la datele obţinute în afara procesului (prin analize de laborator, urmărirea planului de fabricaţie etc.). În sfîrşit, unele semnale prelucrate de procesor pot fi de provenienţă „in-line", adică prelevate (extrase) din memoria internă sau din fişierele memoriei externe (aceste semnale reprezintă valori constante, date programate iniţial, date de referinţă etc.).

După cum s-a mai arătat (v. subcapitolul 10.2), din punctul de vedere al formei lor, semnalele prelevate pot fi: analogice sau numerice („digitale"). Semnalele analogice sînt fie standardizate (ca nivele valorice), fie nestandardizate. Semnalele numerice, după felul senzorilor de la care provin, pot fi: de tip contact sau sub forma unui tren de pulsuri (de tensiune electrică), modulat sau codificat.

Senzori analogici. Aceştia sînt, de fapt, traductoare electrice (sau şi de altă natură, însoţite însă de un convertor cu mărime de ieşire electrică),

Scanned with OKEN Scanner

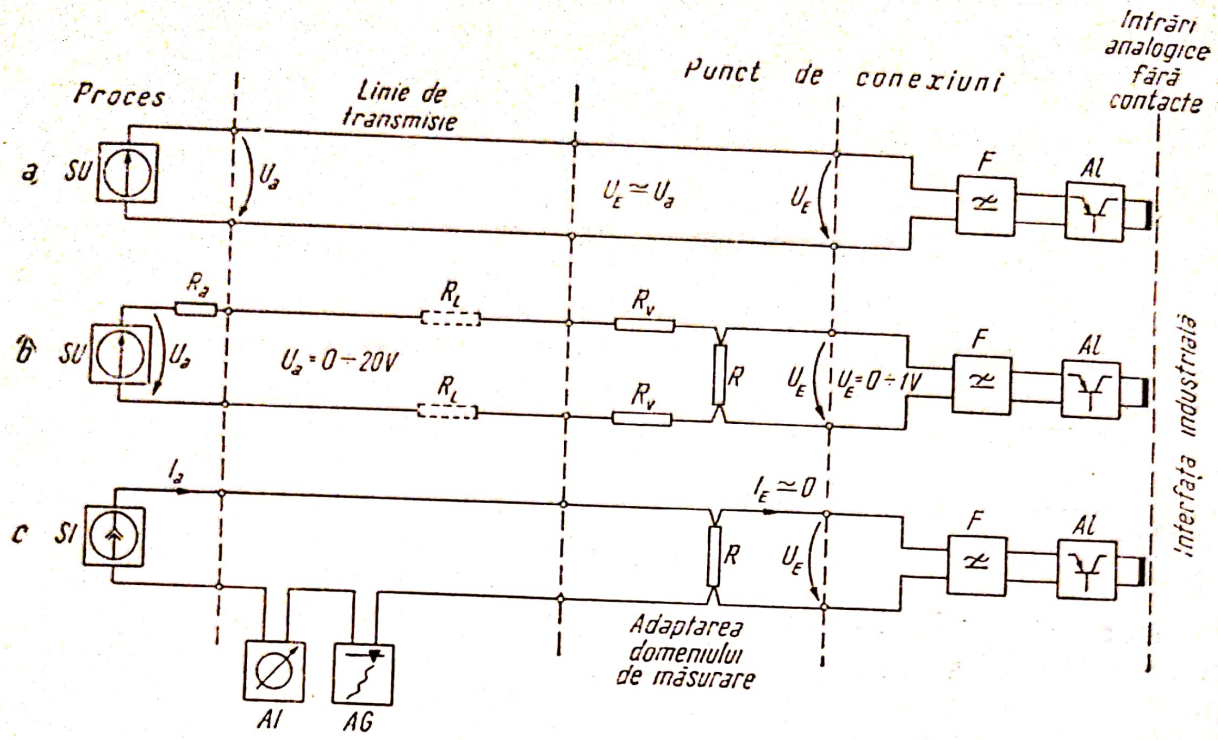


Fig. 11.14.

care produc tensiuni sau curenţi electrici cu valori continue în timp, proporţionale cu parametrii măsuraţi. Traductoarele electrice, după cum se ştie, pot fi de tip parametric sau de tip generator (de curent continuu sau de curent alternativ).

Senzori de curent continuu. Aceştia emit semnale analogice sub forma unor tensiuni electrice continue sau curenţi electrici continui, cu valori limită ale domeniului de măsurat: 10 mV sau 20 V (pentru tensiuni) şi 20 mA (pentru curenţi). Aceşti senzori sînt conectaţi la intrările analogice ale interfeţei industriale aşa cum este arătat în schemele din figura 11.14, în care: *SI* sînt senzori de curent; *SU* — senzori de tensiune; *AI* este ampermetru cu ac indicator; *AG* — ampermetru cu înregistrare grafică; R_a — rezistor adiţional; R_v — rezistor serie pentru adaptarea semnalului de tensiune; *R* — potenţiometru (rezistor paralel); *F* — filtru trece jos; *AL* — adaptor de linie pentru cuplarea directă, fără contacte, de genul repetorului pe emitor, la intrarea analogică a interfeţei industriale. Rezistorul desenat punctat, R_L, reprezintă rezistenţa proprie a liniei de transmisie a semnalului de la senzori la interfaţă.

Aşa cum rezultă din figura 11.14, *a*, senzorii cu tensiunea la borne U_a cu valoarea de lucru corespunzătoare tensiunii nominale (proiectate) a intrării analogice sînt conectaţi direct, printr-o simplă linie bifilară de transmisie. Senzorii de tensiune cu limite mari ale domeniului de măsurare (de exemplu cu 20 V) sînt conectaţi prin rezistoare adiţionale (R_a) şi prin divizoare de tensiune (R_v şi *R*), aşa ca în figura 11.14, *b*. Divizorul de tensiune este necesar pentru reducerea tensiunii de la bornele intrării analogice (U_E) la valoarea maximă permisă la aceste intrări. În cazul senzorilor de curent (fig. 11.14, *c*), curenţii continui produşi de senzori

Scanned with OKEN Scanner

sînt transformaţi în tensiuni continue prin căderea de tensiune dată de curentul I_a printr-un rezistor paralel R.

Schema din figura 11.14 reprezintă cazul cuplării senzorilor analogici pentru măsurarea continuă, deci a valorilor instantanee, prin legături permanente, fără contacte. Dacă nu este necesară măsurarea permanentă (a valorilor instantanee), atunci se folosesc — pentru cuplarea senzorilor la intrările analogice — legăturile cu contacte comandate de un selector (care închid succesiv circuitele de măsurat); în acest fel se obţine, pentru fiecare parametru, valoarea lui medie pe un anumit interval de timp [0].

Problema cea mai importantă, în cazul cuplării senzorilor de curent continuu, este aceea a asigurării tensiunii electrice U la intrările analogice sub o anumită valoare limită maximă admisibilă (de exemplu 1V). După cum am văzut (fig. 11.14), în acest scop se folosesc diverse rezistoare (serie şi paralel); acest fapt poate conduce la erori însemnate dacă rezistenţa proprie a senzorului, a liniei şi a adaptorului domeniului de măsurare (format din rezistoare) nu au o valoare echivalentă corespunzătoare în raport cu rezistenţa proprie (de intrare) a intrărilor analogice. În plus, prezenţa liniei de transmisie (care are o rezistenţă R_L) şi a rezistorilor adiţionali (serie şi paraleli: R_a, R_v şi R) pot produce erori şi prin variaţia propriilor lor rezistenţe, datorită modificării temperaturii mediului ambiant sau apariţiei în semnalul transmis a unor componente variabile în timp cu frecvenţă prea mare, faţă de care rezistenţa echivalentă R_A a dispozitivului analog: senzor — linie — adaptor, şi care este: $R_A = R_a + (2R_L + R_v) + R$ (v. fig. 11.14, *b*), creşte direct proporţional cu frecvenţa.

În cazul tranmisiei directe a semnalului printr-o linie scurtă (v. fig. 11.14, *a*), dacă rezistenţa echivalentă a liniei este foarte mică în raport cu rezistenţa proprie a senzorului, căderea de tensiune pe linie este nesemnificativă (practic $U_E \rightarrow U_a$) şi deci eroarea pricinuită de variaţiile necontrolabile ale acestei căderi de tensiune $U_E - U_a \simeq 0$ sînt neglijabile. Astfel, de exemplu, dacă rezistenţa de intrare a „intrării" analogice este $R_i = 1\text{M}\Omega$ şi domeniul de măsurare este împărţit (eşantionat) de convertorul analog-numeric al interfeţei industriale în 2^{12} trepte, atunci măsurarea suferă o eroare de 0,025% (cel mult un plus din cele 4 096 în care a fost convertit semnalul analog), dacă suma dintre rezistenţa senzorului (R_a) şi rezistenţa liniei ($2R_L$) este $R_A = R_a + 2R_L = 250\Omega$. Mai trebuie avut în vedere şi faptul că rezistenţa dispozitivului analog R_A (senzor-linie de transmisie) scade pronunţat odată cu frecvenţa pulsurilor de explorare a dispozitivelor de măsurare din proces.

Dacă, pentru adaptarea domeniului de măsurare, în circuitul senzorului este inclus şi un divizor de tensiune aşezat la intrarea analogică (v. fig. 11.14, *b*), micşorîndu-se astfel rezistenţa echivalentă R_A, atunci rezistenţa liniei de transmisie nu mai este neglijabilă în raport cu valoarea $R_A = R_a + 2(R_L + R_V) + R$, a întregului circuit exterior faţă de intrarea analogică. În această situaţie, efectul variaţiei temperaturii mediului ambiant asupra rezistenţei liniei $2R_L$ nu mai este neglijabil în orice situaţie. Astfel, de exemplu, pentru ca la o variaţie a temperaturii cu 20 °C eroarea să nu depăşească 0,2%, dacă rezistenţa liniei este $2R_L = 40\ \Omega$, atunci trebuie ca rezistenţa circuitului exterior să nu depăşească 3,4 k Ω ($R_A < 3{,}4$ k Ω şi dacă $R_i = 1\text{M}\,\Omega$). De aceea, pentru corectarea rezistenţei exterioare

Scanned with OKEN Scanner

R_A, în adaptorul domeniului de măsurare se introduc rezistoarele adiționale R_V (v. fig. 11.14, *b*), în limita valorii maxime admise de R_A (de exemplu, 3,4 kΩ), astfel încît $R_L \ll R_V$, ceea ce reduce mult efectul variației rezistenței liniei R_L.

În cazul senzorilor de curent (v. fig. 11.14, *c*), rezistorul R necesar pentru conversia $I_a \rightarrow U_E$ trebuie calculat în funcție de rezistențele liniei de transmisie și a senzorului, dar și ale aparatelor locale de măsurat (de exemplu: AI și/sau AG din figura 11.14, *c*), care sînt conectate în serie. Dacă curentul în linia intrării analogice I_E este foarte mic în raport cu semnalul I_a generat de senzor ($I_E \ll I_a$), atunci valoarea R se alege astfel încît la valoarea maximă a lui I_a să rezulte $U_E = RI_a$ la nivelul maxim admis de intrarea analogică. Dacă valorile lui I_a sînt relativ mici, trbuie luate aceleași precauții în ceea ce privește influența rezistenței liniei de transmisie ca și în cazul senzorilor de tensiune prevăzuți cu divizor rezistiv.

Semnalele generate de senzori sînt, adesea, distorsionate datorită neliniarităților circuitului exterior (în special elementului sensibil al senzorului și eventualelor convertoare montate după senzori) sau datorită unor zgomote care apar în linia de transmisie. Aceste semnale perturbatoare (cu frecvența de 50 Hz și multiplii ei) sînt atenuate (eliminate) de filtrele trece jos montate la intrarea analogică (*F* în figura 11.14).

O categorie aparte de senzori analogici de tip generator de tensiune continuă, foarte utilizați, o constituie *traductoarele cu termocupluri*. Acestea produc tensiuni termoelctromotoare care sînt acceptate direct (ca nivel) de intrările analogice ale interfeței industriale (așa ca în figura 11.14, *a*). Deoarece intrările analogice au rezistențe proprii mari (de exemplu, 1 MΩ), chiar și în domenii mici de măsurare ale termocuplurilor nu sînt posibile falsificări de semnal la baleirea (comutarea) tensiunilor termoelectromotoare, chiar și atunci cînd sînt incluse cabluri de echilibrare cu rezistențe mari. Locul de compensare (comparația cu temperatura mediului ambiant) pentru mai multe termocpuluri (adesea diferite) îl constituie un așa-numit predivizor, unde sînt conectate sfîrșitul cablurilor de echilibrare (fig. 11.15). Compensarea, față de temperatura locului de comparație, se face prin intermediul programului de prelucrare a datelor rezultate din măsurători, în care temperatura de comparație este fie o constantă (dacă temperatura

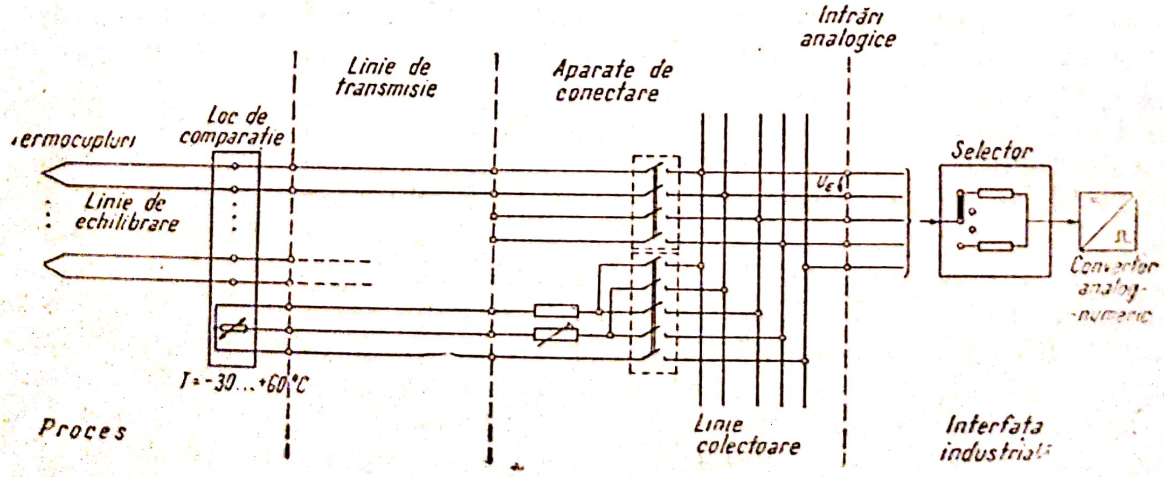


Fig. 11.15.

locului de comparaţie este menţinută constantă cu ajutorul unui termostat), fie o valoare determinată separat prin măsurători (cu un traductor termorezistiv conectat la o intrare analogică suplimentară). În cazul folosirii unui termostat, funcţionarea acestuia trebuie supravegheată; astfel, dacă termostatul iese din funcţiune se comunică un mesaj de întrerupere calculatorului de proces prin intermediul unei intrări numerice.

Senzori de tip parametric. Cei mai răspîndiţi senzori de tip parametric sînt cei care se bazează pe utilizarea traductoarelor rezistive (ca, de exemplu, termorezistenţa). Elementul sensibil al unui senzor rezistiv este format fie dintr-un fir rezistiv calibrat, fie dintr-un corp semiconductor (sub formă de plăcuţe sau tuburi), a cărui rezistenţă poate varia de la 0 la 100 Ω sau de la 20 la 120 Ω (deci cu un interval de măsurat corespunzător unei variaţii maxime de rezistenţă $\Delta R = 100\ \Omega$).

Variaţia rezistenţei senzorului este transformată într-o variaţie de tensiune; aceasta se poate realiza în două moduri diferite, în ambele utilizîndu-se o sursă auxiliară de curent continuu, cu tensiune la borne U_0 stabilizată.

1° Într-unul din moduri (fig. 11.16, *a*), sursa auxiliară alimentează direct (prin două fire) rezistorul senzorului de la care se transmite căderea de tensiune U_E determinată de curentul de alimentare I_a (produs de sursa cu tensiunea U_0) prin rezistorul senzorului. Dacă curentul din circuitul intrării analogice I_E este neglijabil în raport cu valoarea curentului de alimentare (adică $I_E \ll I_a$), semnalul de tensiune U_E transmis intrării analogice a interfeţei industriale este dat de expresia:

$$U_E = U_0 \frac{R_A + R_x}{R_A + \Delta R + 2R_V},$$

unde, în afara notaţiilor U_0 şi U_E (a căror semnificaţie a fost arătată mai sus), se mai utilizează simbolurile: R_A pentru rezistenţa iniţială a senzorului; ΔR — variaţia instantanee a rezistenţei senzorului; R_x — fracţiunea din ΔR prin care se culege căderea de tensiune şi R_V — rezistenţa unor rezis-

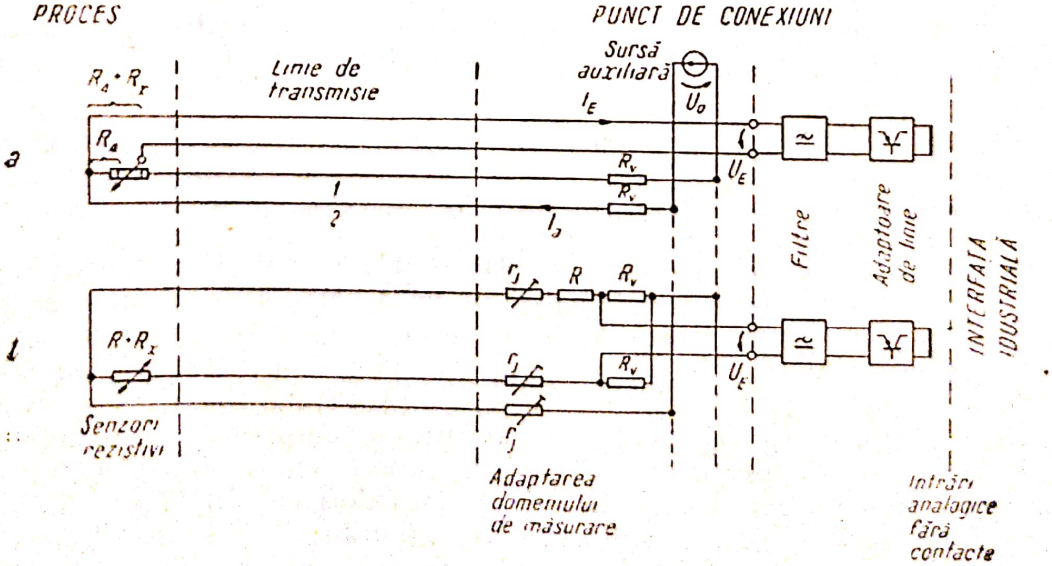


Fig. 11.16.

toare adiţionale (în serie cu senzorul rezistiv), care include şi rezistenţa firelor de alimentare *1* şi *2* (v. fig. 11.16, *a*) foarte mici în raport cu R_V adiţional. Rezistoarele R_V (relativ mari) se introduc pentru a diminua efectul variaţiei de temperatură a mediului asupra rezistenţei conductoarelor de alimentare *1* şi *2* (fig. 11.16, *a*), ceea ce ar influenţa rezultatele măsurătorilor. Datorită valorilor mari ale rezistenţei R_V, tensiunea U_E este de maximum 100 mV (ceea ce face ca la o rezistenţă proprie a intrărilor analogice de cca. 1 MΩ, curentul de intrare I_E să fie efectiv neglijabil). De aceea, intrările analogice ale interfeţei proces-calculator, pentru astfel de senzori, trebuie să aibă o sensibilitate mai mare. Oricum, efectul rezistenţei iniţiale R_A a senzorului şi a rezistenţei conductoarelor de alimentare asupra rezistenţei totale a senzorului $(R_A + \Delta R)$ trebuie avut în vedere şi este cuprins într-o subrutină de corecţie (etalonare) din programele de aplicaţie ale calculatorului de proces. Prin această subrutină se determină două constante de adaptare: factorul $F = (M_E - M_A)/(S_E - S_A)$ şi o valoare auxiliară $B = M_E - F.\ S_E$, unde M_A şi M_E sînt valori proporţionale cu poziţiile iniţiale (indicele A) şi finale (indicile E) determinate pentru senzorul rezistiv, iar S_A şi S_E sînt valorile numerice corespunzătoare obţinute de la convertorul analog-numeric.

2° În celălalt mod, sursa auxiliară stabilizată alimentează o punte Wheatstone, în una din laturile căreia se găseşte senzorul rezistiv cu rezistenţa instantanee $R + R_x$ (unde R este rezistenţa de bază, de repaus sau iniţială a senzorului, iar R_x este variaţia de lucru a rezistenţei senzorului). Acest mod de prelevare a semnalelor este reprezentat în figura 11.16, *b* şi el este specific senzorilor cu traductoare termorezistive sau tensiometrice. Echilibrarea diferenţelor de rezistenţă dintre laturile punţii şi conductorul de alimentare se realizează cu ajutorul unor rezistoare ajustabile la o rezistenţă r_j (v. fig. 11.16, *b*). Semnalul de măsurare este tensiunea U_E de la ieşirea punţii atunci cînd aceasta s-a dezechilibrat prin apariţia variaţiei de rezistenţă R_x, de lucru a senzorului. După cum se vede din figura 11.16, *b*, cînd senzorul este în poziţia iniţială, de repaus, laturile adiacente ale punţii au aceeaşi rezistenţă $R + r_j$ sau R_V, astfel că se realizează condiţia de echilibru a punţii $(R + r_j)R_V = R_V(R + r_j)$. Prin dezechilibrarea punţii, datorită componentei R_x, se obţine tensiunea U_E aplicată intrării analogice:

$$U_E = U_0 \frac{R_x \cdot R_V}{(R_V + R + r_j) \cdot (R_V + R + 3r_j) + r_j R_x},$$

care în general are valori cuprinse între 0 şi 100 mV. Rezistenţa R_V, din laturile neactive ale punţii, determină pe de o parte sensibilitatea secţiunii de măsurat, făcînd posibilă adaptarea domeniului mărimii de măsurat la domeniul de tensiune necesar la intrarea analogică a interfeţei industriale. Pe de altă parte, prin valoarea R_V sînt influenţate liniaritatea secţiunii de măsurat şi eroarea provocată de variaţia cu temperatura a rezistenţelor conductoarelor de alimentare. În cazul ruperii cablului sau deteriorării senzorului etc., semnalul de măsurat U_E ia valori care depăşesc limita de lucru. Pentru acest caz, o subrutină de „tendinţe" şi „plauzibilitate" a programului de prelucrare a datelor detectează această stare (defecţiune). O altă subrutină a programului de prelucrare a valorilor măsurate face

Scanned with OKEN Scanner

posibilă eliminarea neliniarităţilor determinate de senzor şi rezistorul adiţional R_v, prin reetalonări corespunzătoare.

Senzori de curent alternativ. Sînt utilizaţi pentru supravegherea proceselor electrice care au loc în instalaţiile electroenergetice, electromecanice, electrochimice ş.a. de curent alternativ (ca de exemplu: alternatoarele din centralele electrice, staţiile de transformatoare, liniile de transport şi reţelele de distribuţie a energiei electromagnetice, motoarele electrice şi grupurile de maşini electrice pentru acţionări, instalaţii de sudare electrice, cuptoare electrice ş.a.m.d.), care lucrează în „curenţi tari" (puteri mari, tensiuni electrice ridicate etc.). În aceste cazuri, senzorii sînt de fapt transformatoare electrice de măsură (de curent şi de tensiune). Semnalele „generate" de aceste transformatoare sînt mărimi electrice alternative, cu formă de undă practic sinusoidală, cu frecvenţa de 50 Hz şi cu valori eficace cuprinse în domeniul de măsurare standardizat pentru transformatoarele de măsură, adică: curenţi de $0 \div 5$ A (uneori $0 \div 1$ A) şi tensiuni $0 \div 100$ V. În cazul instalaţiilor de joasă tensiune (120 V, 127 V, 208 V, 220 V şi 380 V) se poate renunţa la transformatoarele de tensiune şi atunci semnalele referitoare la tensiunile electrice pot avea valorile eficace de la 100 V la 380 V; totuşi, pentru securitatea intrărilor în interfeţele proces-calculator, prelevarea acestor tensiuni nu se face prin legături directe de conducţie („galvanice"), ci prin cuplaje inductive (cu transformatoare de separare).

Semnalele produse de transformatoarele de măsură sînt transmise prin conductoare la punctul de conexiune (v. fig. 11.13), unde sînt redresate (cu punţi cu diode semiconductoare) şi atenuate (cu divizoare de tensiune) la domeniul de măsurare a intrărilor analogice (de obicei $0 \div 1$ Vc.c. sau uneori: $0 \div 5$ Vc.c., $0 \div 10$ Vc.c., $-1 \div 1$ Vc.c., $1 \div 5$ Vc.c. etc.). Semnalele de curent sînt transformate, prin rezistoare potenţiometrice, în căderi de tensiune.

Înainte de a fi „introduse" în interfaţa industrială, tensiunile redresate (care au o formă pulsatorie) sînt filtrate (printr-un filtru „trece jos") pentru extragerea componentei care va fi, la o anumită scară, o măsură a tensiunilor şi curenţilor electrici din instalaţia condusă.

Deoarece adaptarea valorii prelevate din proces la domeniul de măsurare al intrărilor analogice din interfaţa industrială a calculatoarelor de proces este de importanţă hotărîtoare pentru conducerea proceselor, încheiem acest paragraf (cu privire la senzorii analogici) cu un exemplu referitor la instrumentaţia care poate fi utilizată pe scară largă, în momentul de faţă, în ţara noastră.

Ex. 11—5. La elaborarea dispozitivelor de cuplare la proces (*DIA*, *DIN* şi *DENA* — v. fig. 11.3) ale calculatorului de proces FELIX C.—32P (v. *Ex. 11—1*) s-a avut în vedere — în primul rînd — necesitatea realizării compatibilităţii de conectare a acestor dispozitive cu elementele de automatizare convenţională utilizate mai frecvent în ţara noastră. Spre exemplificare, prezentăm gradul de compatibilitate, referitor la semnalele analogice, al calculatorului de proces FELIX C—32P, prin cuplorul său, cu: sistemul E—LINE produs de IEA (tabela 11—1); sistemul nou de reglare, în curs de asimilare la IEA pe baza concepţiei IPA (tabela 11—2) şi sistemul universal internaţional de reglare şi conducere SRU (tabela 11—3).

În afara domeniilor de lucru indicate în aceste tabele, calculatorul de proces FELIX C—32P asigură în mod suplimentar, intrări analogice

Scanned with OKEN Scanner

Tabela 11—1

Compatibilitatea calculatorului de proces FELIX C—32P cu sistemul E—LINE

Natura semnalului	Domeniul de măsurare (gama semnalului)			
	la intrarea în calculator		la ieşirea din calculator	
	ieşiri elemente sistem	intrări cuplor	intrări elemente sistem	ieşiri cuplor
1	2	3	4	5
Semnale unificate de curent	2 la 10 mA c.c. [1]	da	2 la 10 mA c.c. [1]	da
	0 la 20 mA c.c.	da	0 la 20 mA c.c.	nu [2]
	0 la 5 mA c.c.	da	0 la 5 mA c.c.	da
	1 la 5 mA c.c.	da	1 la 5 mA c.c.	nu [2]
	4 la 20 mA c.c.	da	4 la 20 mA c.c.	nu [2]
	0 la 10 mA c.c.	da	0 la 10 mA c.c.	da
	0 la 50 mA c.c.	da	0 la 50 mA c.c.	nu [2]
	0,5 la 5 mA c.c.	da	0,5 la 5 mA c.c.	nu [2]
	10 la 50 mA c.c.	da	10 la 50 mA c.c.	nu [2]
Semnale unificate de tensiune	0 la V 5 c.c.	da	0 la 5 V c.c.	da
	1 la 5 V c.c.	da	1 la 5 V c.c.	da
	0 la 10 V c.c.	da	0 la 10 V c.c.	da
	0,4 la 2 V c.c.	da		
Semnele neunificate de tensiune	0 la 1 V c.c. $R_i = 50\Omega/mV < 1k\Omega$	da		
	0 la 5 mV	da		

Tabela 11—2

Compatibilitatea calculatorului de proces FELIX C — 32P cu sistemul nou de reglare (IPA)

1	2	3	4	5
Semnale unificate de curent	4 la 20 mA c.c.	da	4 la 20 mA c.c.	nu
	0 la 5 mA c.c.	da	0 la 5 mA c.c.	da
	0 la 10 mA c.c.	da	0 la 10 mA c.c.	da
	0 la 20 mA c.c.	da	0 la 20 mA c.c.	nu
	1 la 5 mA c.c.	da	1 la 5 mA c.c.	da
	2 la 10 mA c.c.	da	2 la 10 mA c.c.	da
Semnale unificate de tensiune	0 la 5 V c.c.	da	0 la 5 V c.c.	da
	0 la 10 V c.c.	da	0 la 10 V c.c.	da
	1 la 5 V. c.c.	da	1 la 5 V c.c.	da

Tabela 11—3

Compatibilitatea calculatorului de proces FELIX C—32P cu sistemul SRU

1	2	3	4	5
Semnale de curent	0 la 5 mA c.c.[1]	da	0 la 5 mA c.c.[1]	da
	−5 la +5 mA c.c.	da[4]	−5 la +5 mA c.c.	da
	0 la 20 mA c.c.	da	0 la 20 mA c.c.	nu
	−20 la +20 mA c.c.	da[4]	−20 la +20 mA c.c.	nu
	0 la 100 mA c.c.	da	0 la 100 mA c.c.[3]	nu
	−100 la +100 mA c.c.[3]	da[4]	−100 la +100 mA c.c.[3]	nu
Semnale de tensiune	0 la 1 V c.c.	da	0 la 1 V c.c.	da
	−1 la +1 V c.c.	da[4]	−1 la +1 V c.c.	da[4]
	0 la 10 V c.c.[1]	da	0 la 10 V c.c.[1]	da
	−10 la +10 V c.c.	da[4]	−10 la +10 V c.c.	da[4]

[1] Domeniu preferenţial.
[2] Domeniu utilizat numai la unele instalaţii mai vechi.
[3] Domeniu utilizat numai prin înţelegere prealabilă între beneficiar şi producător.
[4] Domeniu acceptat numai prin comandă specială.

Scanned with OKEN Scanner

pe tensiune cu următoarele game: 0 la 10 mV, 0 la 20 mV, 0 la 40 mV, 0 la 50 mV, 0 la 100 mV, 0 la 200 mV, 0 la 400 mV, 0 la 500 mV și 0 la 2 V (toate în curent continuu).

Fabricantul calculatorului FELIX C—32P (Intreprinderea de calculatoare electronice București) poate rezolva — la cererea utilizatorilor — și alte probleme de compatibilitate (adaptarea rezistențelor de intrare cu rezistențele circuitelor exterioare, lungimea transmisiei, filtrarea semnalelor etc.). De exemplu, pentru toate semnalele extrase din proces sînt prevăzute, opțional, și elemente de filtrare.

Senzori numerici (digitali). În majoritatea instalațiilor ale căror procese sînt conduse cu calculatoare există și numeroase aparate de măsurat (din sistemele de automatizare convenționale, dar mai ales speciale) ce produc semnale numerice care — codificat — exprimă valoarea parametrilor procesului. Aceste semnale numerice sînt transmise calculatorului prin „intrările" numerice (digitale) ale interfeței industriale (proces — calculator).

Așa cum am mai arătat (v. pag. 129), aceste semnale numerice pot fi:

1) de tip contact (de exemplu: contactele de reavizare a poziției supapelor și sertarelor de blocare, contactele limitatoarelor pentru „sfîrșit de cursă", contactele de indicare a funcționării unui dispozitiv tehnologic prin „DA sau NU", contactele de semnalizare a „plinului" sau „golului" în rezervare etc.) cu semnale în formă de treaptă de tensiune;

2) trenuri de pulsuri, modulate sau codificate, provenite de la traductoarele numerice sau de la ieșirea convertoarelor analog- numerice ale instrumentației de proces, aparatelor de codare, claviaturilor etc.

Semnalele numerice se deosebesc între ele prin: conținutul informației, sistemul de codificare și „comportarea" în timp (potrivit căreia unele semnale numerice sînt dirijate la așa-numitele „intrări numerice statice", iar altele la „intrările numerice dinamice" ale interfeței industriale). Semnalele referitoare la proces, produse de senzorii numerici, după ce sînt transmise la intrările numerice ale interfeței industriale, sînt preluate de calculator și ori sînt imediat prelucrate (în timp real) conform unor programe, ori sînt stocate în memoria externă în vederea unor ulterioare prelucrări.

În ceea ce privește intrările numerice ale interfeței industriale, acestea pot fi de două tipuri: statice și dinamice.

Intrările numerice statice sînt folosite atunci cînd semnalele numerice prelevate din proces prezintă interes față de anumite momente sau evenimente de referință (definite, deci „statice") și introduse în calculator prin program, putînd declanșa — în funcție de conținutul informațional al semnalului numeric sincronizat cu stările de referință din program — anumite prelucrări de date și operări (comenzi pentru proces). Un exemplu ar fi acela al semnalelor numerice cu privire la poziția unor organe ale instalației conduse: supape, sertare, robinete etc. a căror „referință" este poziția de sfîrșit, de blocare, de închidere etc.; în funcție de conținutul semnalelor numerice de reavizare a acestor organe (semnale ce sînt introduse prin intrările numerice statice), calculatorul care conduce procesul elaborează — conform unui program — comenzi pentru acționarea elementelor de execuție ale dispozitivelor de automatizare a procesului.

Scanned with OKEN Scanner

Intrările numerice dinamice servesc la recunoaşterea modificărilor de stare a semnalelor binare individuale, adică de la fiecare poziţie („bit cu bit"), faţă de o stare dată anterior, precum şi la înmagazinarea în memorii tampon sau externe a informaţiei semnalelor sub formă de trenuri de pulsuri, pînă la preluarea ei de către memoria operativă (internă, a unităţii de prelucrare). Deci, intrările numerice dinamice realizează — pe lîngă recunoaşterea selectivă a modificării unuia sau mai multor bits ai semnalului numeric — şi memorarea de scurtă durată (între două procese de calcul sau operaţii consecutive) a unor valori codificate în binar, cu una sau mai multe poziţii.

Senzori numerici de poziţie. Elementele de ieşire ale acestor senzori, care generează semnalele sub forma unor trepte de tensiune electrică, sînt fie contacte mecanice fie circuite de comutaţie cu tranzistoare (cu două ieşiri cu potenţiale distincte — ieşirea simetrică, sau cu o singură ieşire faţă de masa comună — adică ieşirea asimetrică).

Aceşti senzori se folosesc pentru indicarea valorilor limită ale unor parametri ai procesului, pentru avertizarea depăşirii unor valori maxime sau minime admisibile, a unor stări de avarie, de pericol etc., ale apariţiei unor perturbaţii inadmisibile, ale unor manevre greşite ş.a.m.d., care — de obicei — impun transmiterea la calculatorul de proces a unor semnale pentru întreruperi externe.

De aceea semnalele numerice generate de senzorii de poziţie (stare) sînt, majoritatea, introduse în interfeţele industriale prin aşa-numitele „intrări de alarmă". *Intrările de alarmă* primesc semnale binare care determină prelucrări cu prioritate atunci cînd aceste semnale „anunţă", prin conţinutul lor informaţional, modificări şi fenomene imprevizibile (anormale) în proces. La intrările de alarmă statice se conectează mai ales senzorii de semnalizare a depăşirii valorilor limită ale unor parametri importanţi ai procesului, a pericolelor, a avariilor şi a altor situaţii de acest gen. La intrările de alarmă dinamice sînt introduse semnalele de explorare periodică (testare) ale stării anumitor fenomene sau instalaţii de bază din proces, precum şi semnalele comunicate de la claviaturile consolelor operatorului tehnolog.

Senzorii de poziţie (cu contacte sau circuite de comutaţie cu tranzistoare) se conectează direct la intrările numerice, prin conductoare (cabluri). Influenţa rezistenţei acestor linii de transmisie este, în general, neglijabilă.

Trecerea semnalului de la intrarea numerică, de la un nivel de tensiune electrică la altul (corespunzător unei stări sau parametru cu valoare binară), trebuie făcută printr-un salt relativ mare de tensiune sau de curent electric pentru a se elimina efectul perturbaţiilor şi — mai ales — pentru a se asigura „străpungerea" sigură a straturilor oxidante, care au rezistenţă electrică mare, ce apar la contacte prin coroziunea inerentă a suprafeţelor de contact. De mare importanţă este şi timpul de întîrziere în transmiterea unui semnal; acesta este timpul scurs de la modificarea de tip treaptă a tensiunii de la ieşirea senzorului (cu valoarea nivelului de semnal util) pînă la apariţia unei modificări corespunzătoare a semnalului de la intrarea numerică a interfeţei industriale. Timpul de întîrziere este determinat de structura şi proprietăţile filtrului de la intrarea numerică, de capacitatea proprie a liniei de transmisie, de rezistenţa circuitului de intrare, precum şi de însăşi valoarea treptei de tensiune a semnalului util. Deoarece necesitatea

Scanned with OKEN Scanner

producerii unei trepte relativ mari de tensiune este în contradicţie cu necesitatea unui timp de întîrziere cît mai mic, aceste valori (de nivel util şi timp de transmisie) sînt diferite de la o intrare la alta, chiar în cazul intrărilor numerice de acelaşi tip, şi ele se reglează în mod adecvat în perioada de instalare şi testare a calculatorului de proces. Oricît de atent ar fi realizate aceste reglaje, nu este exclus ca în multe cazuri să apară, pentru scurt timp, un puls perturbator care să falsifice valoarea semnalului numeric obţinut prin codificări în mai multe locuri. Această valoare falsă are efecte eronate atunci cînd, exact în intervalul apariţiei semnalului perturbator, este apelată intrarea numerică de către unitatea de prelucrare. Pentru evitarea unor astfel de erori este necesar fie să se întîrzie procesul de operare (prelucrare), fie — mai bine — să se repete operarea de mai multe ori, la intervale de timp corespunzătoare timpului de întîrziere (transmisie), după care să se compare valorile astfel obţinute. Prin operare repetată pot fi înlăturate efectele perturbaţiilor induse, de scurtă durată. Tot în scopul eliminării unor astfel de erori, se pot folosi şi coduri redundante (cu poziţii suplimentare de validare a datelor), care sînt eficiente în special pentru eliminarea erorilor de natură statică (sistematică).

*Traductoare de pulsuri**. După cum se ştie [10], aparatura de automatizare convenţională include şi numeroase traductoare de pulsuri (ca, de exemplu, numărătoarele de obiecte, numărătoarele de intervale de timp, unele debitmetre, unele tahogeneratoare, unele dozatoare ş.a.) care, la ieşire, generează semnale electrice (tensiuni şi curenţi) de formă pulsatorie. Valoarea parametrului măsurat este exprimată de semnalul pulsatoriu printr-un procedeu de modulaţie: în amplitudine, în frecvenţă, în durată sau în fază (poziţie) [22].

Prelucrarea unor astfel de semnale pulsatorii de către calculatorul de proces pune numeroase probleme de prelevare şi interpretare printre care principalele sînt analizate (sumar) în continuare.

1° Demodularea semnalului pulsator care, de fapt, se reduce la discriminarea a două situaţii admise: numărarea de pulsuri limitată de două evenimente din decursul procesului (ca, de exemplu, în cazul măsurării obiectelor) şi determinarea frecvenţei pulsurilor, adică numărarea pulsurilor produse în unitatea de timp (ca, de exemplu, în cazul măsurării debitelor de fluid cu un traductor de frecvenţă). În acest al doilea caz (măsurarea frecvenţei pulsurilor), trebuie precizată în prealabil baza de timp, ceea ce se realizează cu un „timer".

2° Asigurarea calităţii semnalului extras prin senzorii de proces, atît ca formă de undă (de obicei dreptunghiulară, uneori însă sinusoidală) cît şi ca amplitudine (valoare maximă). Adeseori, semnalele produse de senzorii din proces trebuie „reformate", adică: amplificate ca nivel de tensiune; filtrate, pentru eliminarea perturbaţiilor produse prin distorsiuni (în special cele datorite neliniarităţii traductoarelor); supuse unei conversii de cod pentru a corespunde sistemului de codificare admis la intrările numerice etc. În acest ultim scop, de cele mai multe ori, se utilizează un formator de pulsuri (*FP* pe schemele din figurile: 11.17, 11.18 şi 11.19, ce vor

* Cititorul a remarcat, desigur, opţiunea noastră pentru termenul de *puls* (în loc de *impuls*), fiind vorba de fapt de un semnal fizic a cărui mărime (valoare) este periodică fără a-şi schimba semnul (v. *Lexiconul tehnic român* — vol. 10 Bucureşti, Editura tehnică, 1962 — pag. 612).

Scanned with OKEN Scanner

fi comentate mai jos). În ceea ce privește filtrele (v. *F* în figurile: 11.17, 11.18 și 11.19), care sînt filtre „trece jos", ele trebuie să aibă frecvența superioară, de blocare, cu puțin mai mare decît limitele domeniului frecvențelor de lucru ale traductoarelor de frecvență (de la cîteva cicluri pe minut la cîțiva kilohertz).

3° Introducerea semnalelor de frecvență în interfețele industriale se poate face în mai multe moduri, în funcție de frecvența semnalului prelevat din proces. Astfel:

a) dacă frecvența pulsurilor semnalului de măsurat se află sub o anumită valoare limită inferioară (determinată de încărcarea calculatorului), atunci este posibilă introducerea directă a pulsurilor printr-o intrare numerică cu control direct (prin program) sau prin întreruperi externe. În cazul controlului direct sînt folosite intrările numerice dinamice, care sînt apelate cu o frecvență ce trebuie să fie mai mare decît cea mai mare dintre frecvențele semnalelor ce pot fi produse de senzorii din proces. În această situație, fiecărei intrări numerice îi este repartizată o celulă de memorare, de la care programul de control direct ia, după fiecare puls de numărare recunoscut, valoarea lui binară actuală, o mărește cu 1 și o înmagazinează, din nou, în aceeași celulă de memorare. Valoarea înregistrată reprezintă valoarea de plecare pentru măsurarea pulsurilor din semnal (ce reprezintă, în proces, obiecte, intervale etc.) sau a frecvenței acestor pulsuri;

b) în cazul semnalelor de măsurare cu frecvențe mai mari se folosesc niște contoare auxiliare (v. *NA* din figurile: 11.17, 11.18 și 11.19). Acestea sînt, de fapt, numărătoare binară cu $n+1$ ieșiri paralele (cîte una pentru fiecare poziție valorică a numărului binar) care sînt conectate la intrările numerice statice (v. figurile: 11.17, 11.18 și 11.19). Capacitatea K a fiecărui numărător auxiliar trebuie să fie astfel încît, atunci cînd frecvența pulsurilor semnalului prelevat din proces este maximă, în timpul dintre două înregistrări de valori la intrarea numerică, să se producă maximum $K-1$ pulsuri. Dacă notăm cu Z_{i-1} și Z_i starea numărătorului auxiliar la două momente de tact succesive $i-1$ și i, atunci numărul de pulsuri ΔZ numărat între cele două momente de tact se calculează astfel;

— dacă $Z_i > Z_{i-1} \Rightarrow \Delta Z = Z_i - Z_{i-1}$;

— dacă $Z_i < Z_{i-1} \Rightarrow \Delta Z = Z_i - Z_{i-1} + K$.

În cazul în care aceste contoare auxiliare trebuie operate chiar în timpul numărării pulsurilor (ca, de exemplu, la măsurarea frecvenței semnalului prelevat din proces) este necesar un dispozitiv suplimentar alcătuit dintr-un element de întîrziere și un circuit basculant care să producă, pentru sincronizare, un așa-numit bit de blocare, conform figurii 11.17 în care

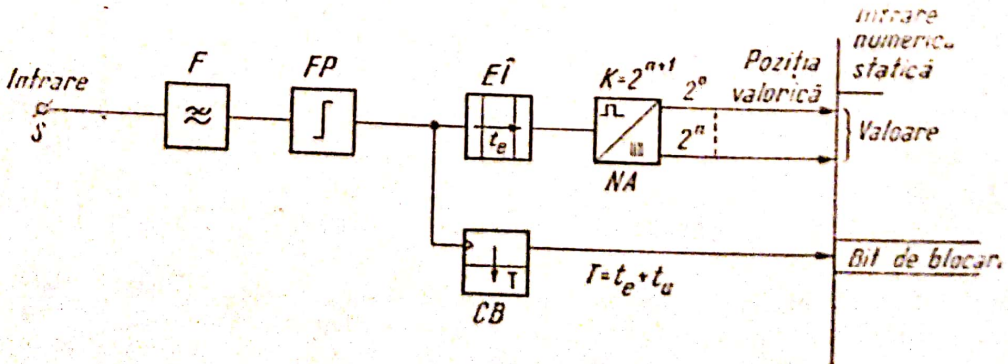


Fig. 11.17.

este reprezentată schema bloc (principială) a unei intrări numerice statice prevăzută cu numărător auxiliar şi dispozitiv suplimentar de comandă. Notaţiile folosite în schema din figura 11.17 au următoarele semnificaţii: S — punctul de intrare în interfaţa industrială a semnalului numeric pulsator prelevat din proces; F — filtru trece jos; FP — formator de pulsuri; NA — numărător auxiliar cu capacitatea $K = 2^{n+1}$; EI — element de întîrziere (cu t_e — timpul de reglare necesar intrării numerice); CB — circuit de basculare, cu timpul de basculare $T = t_e + t_u$, unde t_u este timpul de comutare al numărătorului auxiliar.

Prin această schemă se asigură sincronizarea necesară între operaţiile de preluare a semnalului şi de numărare, prin contorul auxiliar, a frecvenţei acestui semnal. Astfel, bitul de blocare — care apare după un timp T, format din timpul necesar reglării intrării numerice (t_e) şi timpul necesar comutării numărătorului auxiliar (t_u) — provoacă comutarea numărătorului atunci cînd intervine un nou puls de numărat. Prin controlul direct al calculatorului (control programat), se verifică dacă pentru fiecare valoare ce urmează a fi preluată (introdusă în procesor) există bitul de blocare (v. fig. 11.17); dacă bitul de blocare este zero, preluarea valorii din intrarea numerică statică este întîrziată pînă la apariţia acestui bit. În acest fel sînt eliminate erorile determinate de procesele de comutare şi de timpii de întîrziere la transmisia semnalelor. Asemenea erori mai pot fi evitate prin însăşi programul de control (al controlului direct), dacă acesta este prevăzut să comande numărătorul auxiliar de mai multe ori consecutiv, la intervale corespunzătoare timpilor de intrare în oscilaţii proprii ale elementelor din circuitul exterior de măsurare, după care compară valorile preluate, introducînd-o în procesor numai pe cea validată.

Numărătoarele auxiliare pot fi folosite şi ca divizoare de frecvenţă, într-un montaj „combinat" prin care ieşirile paralele ale numărătorului sînt conectate atît la intrările numerice statice cît şi la intrările numerice dinamice, aşa cum se arată în figura 11.18 (notaţiile de aici au aceeaşi semnificaţie ca cele folosite şi în figura 11.17). În montajul din figura 11.18 frecvenţa pulsurilor este foarte mult demultiplicată; astfel, frecvenţa pulsurilor pe linia de ieşire de rang n (notată cu f_u) este de K ori mai mică decît frecvenţa pulsurilor f_i de la intrarea în numărătorul auxiliar, K fiind capacitatea acestui numărător ($f_u = f_i/K$). În acest fel, este posibil ca programul de control direct să poată analiza pulsul transferat la poziţia cea mai semnificativă a valorii preluate, ceea ce se realizează printr-o celulă a intrării numerice dinamice. Pentru comunicarea valorii exacte, după apariţia la intrarea dinamică a bitului de la poziţia n, programul de control direct

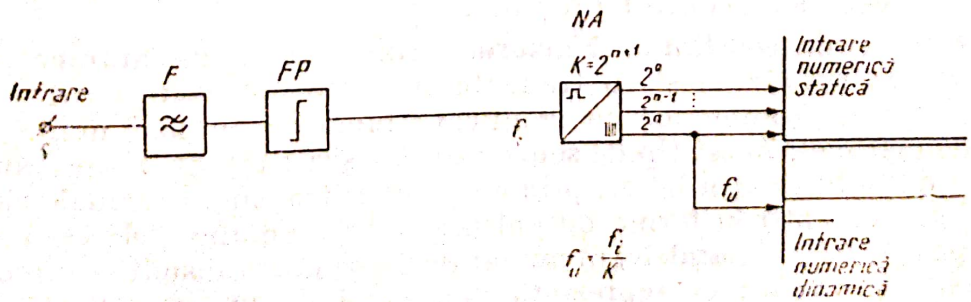


Fig. 11.18.

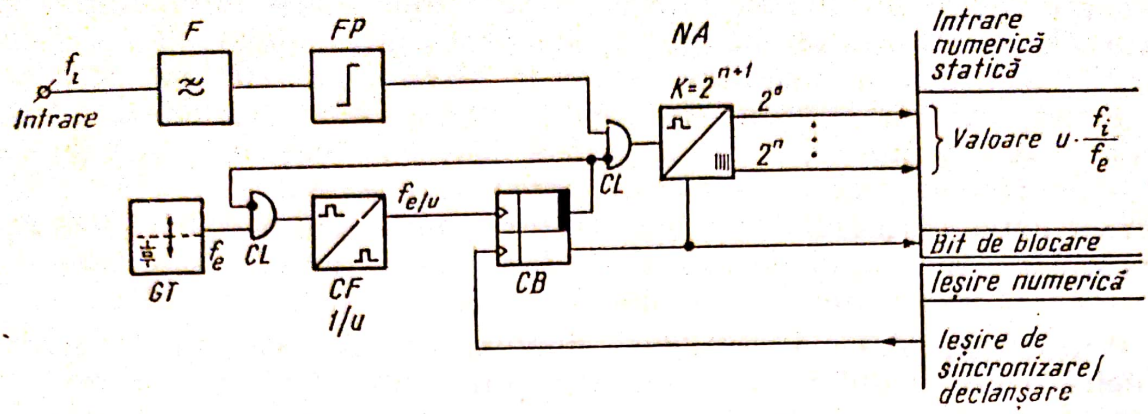


Fig. 11.19.

comandă preluarea celor $n + 1$ ieşiri paralele ale numărătorului auxiliar prin intrarea numerică statică. O astfel de numărare combinată a pulsurilor semnalului de proces are avantajul reducerii elementelor structurale ale fiecărei intrări numerice a interfeţei; în schimb, „complică" întrucîtva programul de control direct.

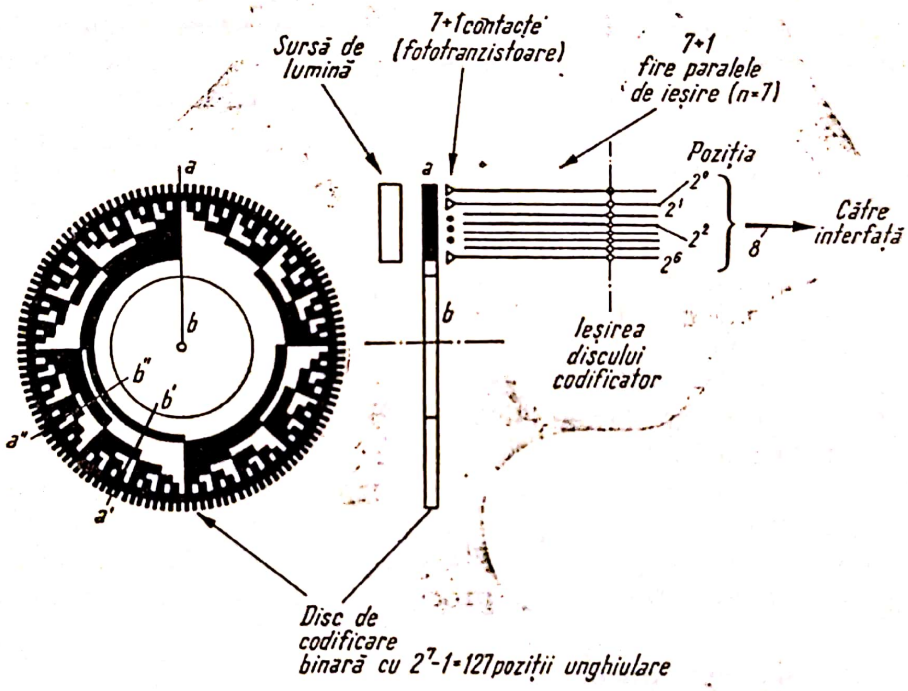
Pentru măsurarea frecvenţei semnalului prelevat din proces, pe lîngă numărătorul auxiliar mai este necesară şi o bază de timp faţă de care valoarea frecvenţei se determină ca fiind numărul de pulsuri ale semnalului de intrare existente în intervalul de timp. Schema de principiu (bloc) pentru o intrare numerică destinată măsurării frecvenţei semnalului de proces este redată în figura 11.19 în care: F este un filtru „trece jos"; FP — formator de pulsuri; NA — numărător auxiliar cu capacitatea K; GT — generatorul bazei de timp; CB — circuit basculat; CL — circuite logice de interdicţie (produc un răspuns numai în cazul existenţei semnalului la intrarea cu punct); CF — convertor de frecvenţă (divizor cu raportul u); f_i — frecvenţa semnalului de intrare (extras din proces); f_e — frecvenţa semnalului ce constituie baza de timp. După cum se vede din schema prezentată în figura 11.19, se măsoară tot numărul de pulsuri ale semnalului de intrare, însă se ia valoarea lui într-un interval de timp $1/f_e$, multiplicat de u ori, deci se măsoară frecvenţa raportată u (f_i/f_e), ceea ce reprezintă la o anumită scară valoarea parametrului prelevată prin traductorul de frecvenţă ce produce semnale cu frecvenţa f_i. Ca urmare, schema din figura 11.19 reprezintă schema din figura 11.18 (care transmite valoarea: număr de pulsuri), completată cu o schemă de generare a intervalului în care — în mod repetat — să se măsoare numărul de pulsuri, numărarea fiind sincronizată şi declanşată la începutul fiecărui interval de timp u/f_e de o ieşire numerică directă sub control program.

Dispozitive de codificare. Numeroase dispozitive de automatizare a proceselor sînt prevăzute şi cu traductoare dotate la ieşire cu dispozitive convertoare analog-numerice cu codificare binară prin mai multe poziţii (ordine de mărime). Acest tip de senzori au n ieşiri paralele, n fiind numărul de poziţii de codificare în binar, prin care pot fi transmise semnale electrice (tensiuni sau curenţi) în formă de pulsuri. Prin aranjamentele cu repetiţie posibile pe cele n ieşiri paralele, în număr de 2^n, se pot transmite — în diferite coduri binare — valori ce reprezintă parametri de proces sau stări care, prin traductor şi dispozitivul de codare, au provocat diversele aranjamente

Scanned with OKEN Scanner

ale semnalelor binare de pe cele n ieşiri paralele ale dispozitivului. Exemple de astfel de dispozitive sînt convertoarele analog-numerice matriceale sau cunoscutul disc codificator unghiular [10].

Ex. 11—6. Discul codificator unghiular (fig. 11.20) poate fi întîlnit în proces acolo unde sînt de măsurat deplasări unghiulare (mişcări de rotaţie) analogice sau orice alte mărimi (forţe, cupluri de forţe, deplasări liniare etc.) care pot fi transformate printr-un element sensibil la deplasări unghiulare. Corespunzător numărului de poziţii de codificare binară (de exemplu 7 sau 8 poziţii), discul are tot atîtea sectoare circulare concentrice cu alternări de depuneri conductoare şi izolante electric sau decupări prin care, cu ajutorul unor contacte sau fototranzistoare paralele (cîte unu



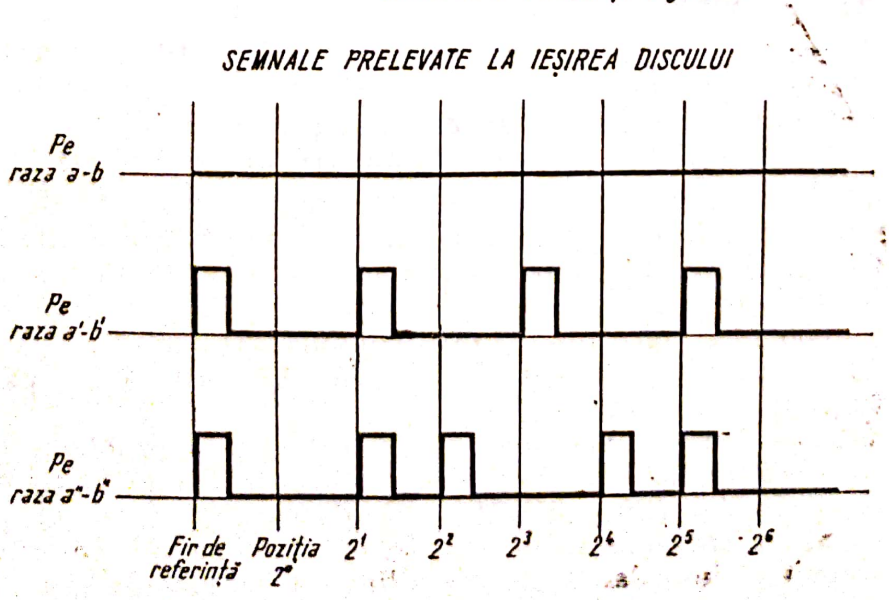


Fig. 11.20.

Scanned with OKEN Scanner

pentru fiecare sector circular și dispuse în lungul razei discului) se pot transmite semnale pulsatorii de tensiune sau curent. În funcție de poziția unghiulară a discului, pe cele n linii se transmit aranjamente distincte de pulsuri care într-un anumit cod binar exprimă poziția discului și prin aceasta valoarea unei mărimi a procesului care a determinat acea poziție.

Ieșirile paralel ale discului de codificare sînt conectate direct la tot atîtea intrări numerice statice ale interfeței proces-calculator.

Determinarea valorii corespunzătoare unui anumit semnal transmis se face de către calculatorul de proces, printr-un program de comparare a structurii pulsurilor din liniile paralele cu un tabel de cod memorat permanent.

În cazul în care prelevarea valorii din proces se face staționar, adică în timpul cît discul nu se rotește ci are o poziție fixă pe durata extragerii semnalelor, codificatorul unghiular indică o valoare precisă, neafectată de erori de trecere a contactului pe disc de la o zonă la alta (de exemplu, de la o zonă de conducție sau transparență optică la alta izolată sau opacă).

În cazul în care ridicarea datelor (semnalului) se face în regim dinamic, adică în timpul rotirii discului, pot apare erori prin semnale false produse de mișcarea acestuia, în momentul trecerii contactelor de la o zonă la alta. Pentru a obține, în astfel de situații, o valoare instantanee nefalsificată există două metode: *1)* reducerea timpului de control direct al intrărilor numerice mult sub durata de rămînere a discului într-o poziție și *2)* eliminarea semnalelor false de comutare — care au o durată mai mică decît cele utile (de contact) — prin utilizarea unui cod descriminator (ca, de pildă, codul Gray cu permutarea ciclică).

Pentru prima metodă, se impune ca frecvența de baleiere, f_t, a intrărilor numerice de către programul de control direct, să respecte condiția: $f_t > V_{max}/\varepsilon$, în care: V_{max} reprezintă viteza de modificare maximă a valorii de măsurat raportată procentual la gama mărimii (în %/secundă) și ε este eroarea admisibilă raportată procentual la gama mărimii considerate.

Transmisia semnalelor

În cadrul acestui paragraf ne vom referi la modalitățile de transmitere a semnalelor electrice, produse de senzorii din proces, pînă la intrările industriale, proces-calculator. Aceste semnale electrice sînt semnale utile, deoarece reprezintă valoarea măsurată a unui parametru al procesului.

În general aceste transmisii se fac pe distanțe relativ scurte, de ordinul a cîteva sute de metri. În cele ce urmează ne vom referi la cazurile în care distanța de transmisie, dintre senzorul din proces și interfața industrială, nu depășește 600 m. Transmisia semnalelor prelevate din proces la distanțe mai mici decît 600 m se face direct, prin conductoare izolate, asociate în cabluri multifilare cu firele răsucite în perechi, blindate și ecranate, cu legături la masă într-un punct comun etc.

Transmisiile pe distanțe mai mari decît 600 m pun probleme dificile de adaptare a impedanțelor și mai ales de echilibrare a impedanței proprii a liniei de transmisie. Așa cum am arătat și în paragraful precedent, pentru ca nivelul semnalului transmis să nu fie atenuat de linie sub valorile unui semnal util, trebuie ca impedanța proprie a liniei de transmisie să fie foarte mică în raport cu impedanțele de adaptare la intrările interfeței (deci față

Scanned with OKEN Scanner

de impedanța de intrare). Această condiție este, însă, greu de satisfăcut în cazul unor cabluri de transmisie mai lungi decît 500 ÷ 700 m. În cazul transmisiilor pe distanțe mai mari decît 700 m, dar mai mici decît 1 500 m, efectul atenuator al liniei poate fi compensat prin introducerea unor amplificatoare suplimentare. Pentru distanțe mai mari decît 1 400 m ÷ 1 500 m, transmisia directă a semnalului nu mai este posibilă din cauza efectului perturbator al zgomotelor, al căror nivel devine comparabil cu nivelul semnalului util (cel prelevat de senzori din proces). De aceea, pentru eliminarea semnalelor perturbatoare, în cazul transmisiunilor pe distanțe mai mari decît 1 500 m, se utilizează un semnal purtător modulat (de obicei în frecvență sau fază) de către semnalul util, de măsurat. Semnalul modulat se poate transmite prin orice mijloc: cabluri speciale (locale) de transmisie, rețelele telefonice existente, cablu hertzian (transmisii radio) etc. Dar, pentru aceasta, în punctele extreme ale transmisiei (senzor-proces și interfață-calculator) sînt necesare dispozitive suplimentare de modulare-demodulare (cunoscutele MODEM-uri)*.

Perturbații. În toate cazurile practice (chiar și în situația unor linii simple formate din conductoare electrice cu lungimi mici, de numai cîțiva metri), transmiterea informației prin canale de telecomunicație nu se poate face fără o oarecare degradare a informației, pricinuită de efectul perturbațiilor.

Perturbațiile, în transmiterea informației prin semnale electrice, sînt așa-numitele *zgomote* care reprezintă mărimi electrice neutile (nu conțin date despre proces), ce apar în canalul de telecomunicație fir sub formă de *paraziți* (semnale „străine" de proces, produse în linii prin fenomene de inducție electromagnetică, prin efecte capacitive în izolații și prin conducție în punctele de contact), fie de *distorsiuni* (componente armonice ale semnalului util deformat prin efecte de neliniaritate, de frecvență etc.).

În cazul transmisiei semnalelor analogice, perturbațiile se datoresc în principal semnalelor parazite, care pot apare atît la senzori cît și în linia de transmisie, precum și în aparatele aferente acestei linii (adaptoare, cutii cu borne de conexiuni, amplificatoare ș.a.).

Senzorii din instalațiile procesului condus sînt montați în locuri cu potențiale electrice față de pămînt net diferite și înconjurați de cîmpuri electromagnetice foarte diverse (produse în special de instalațiile electroenergetice ale procesului); ca urmare, senzorii sînt — permanent — sediul inducției unor semnale parazite. Aceste semnale parazite sînt mai mari la senzorii cu borne de ieșire nesimetrice față de pămînt, chiar dacă acești senzori sau carcasele lor de ecranare și blindare sînt legate la pămînt în locul unde sînt montați, direct sau printr-o rezistență mică. Efectul este și mai mare la liniile de transmisie, care leagă senzorii de interfețe, cu atît mai pronunțat cu cît linia este mai lungă și se învecinează cu alte linii electrice străine, energetice și de telecomunicații. Perturbațiile ajung la nivele mai mari în cazul unor montaje mai neglijente care conduc la situații ca:

— apariția unor cuplaje de conducție prin defecte de izolație (îmbătrînirea izolației, umezirea sau prăfuirea ei, străpungerea izolației), micșora-

* Pentru detalii cu privire la transmisia pe distanțe mari a datelor și la teleprelucrare, sfătuim cititorii să consulte lucrarea [22].

Scanned with OKEN Scanner

rea rezistenţei de izolaţie datorită procesului (ca, de exemplu, conducţia prin materiale refractare de izolaţie a traductoarelor termoelectrice şi a termocuplurilor la temperaturi ridicate sau conducţia printr-un fluid — ca în cazul unui electrod de pH-metru dintr-un rezervor metalic legat la pămînt), precum şi producerea unor tensiuni electromotoare de natură electrochimică la contactele oxidate diferit şi umezite etc.;

— existenţa unor cuplaje capacitive între linia de transmisie a semnalelor la calculator şi reţelele de alimentare cu energie electrică locale, cînd acestea sînt plasate prea apropiat, precum şi între carcasele senzorilor sau ale altor aparate ale canalului de transmisie şi pămînt (v. C_A din fig. 11.21);

— producerea unor cuplaje inductive pronunţate prin defecte de ecranare sau prin amplasări prea apropiate ale cablurilor de transmisie a semnalelor la calculator de alte cabluri şi linii electrice străine (de energie electrică, de telefonie, de semnalizare locală etc.).

De aceea, la realizarea unei transmisii proces-calculator trebuie acordată o atenţie specială: amplasării senzorilor şi liniilor, izolaţiei electrice, ecranărilor şi blindajelor, stării contactelor şi legăturilor la masă (pămînt).

În cazul semnalelor analogice, tensiunile parazite din reţeaua de transmisie pot fi de două categorii (fig. 11.21): tensiuni asimetrice (U_p) şi tensiuni simetrice sau de linie (U_l).

Tensiunile asimetrice sînt paraziţi induşi chiar în circuitul de transmisie, suprapunîndu-se direct peste tensiunile utile U_a (v.fig. 11.21). De aceea ele au un efect perturbator net asupra tensiunii de măsurat U_E, deoarece $U_E = U_a + U_p$, trebuind atenuată cît mai mult.

Tensiunile simetrice sînt paraziţi care apar la capătul liniei de transmisie, la ambele contacte ale intrării analogice, considerate faţă de potenţialul de referinţă al senzorului de tip generator (v. bornele *3* şi *4* din fig.

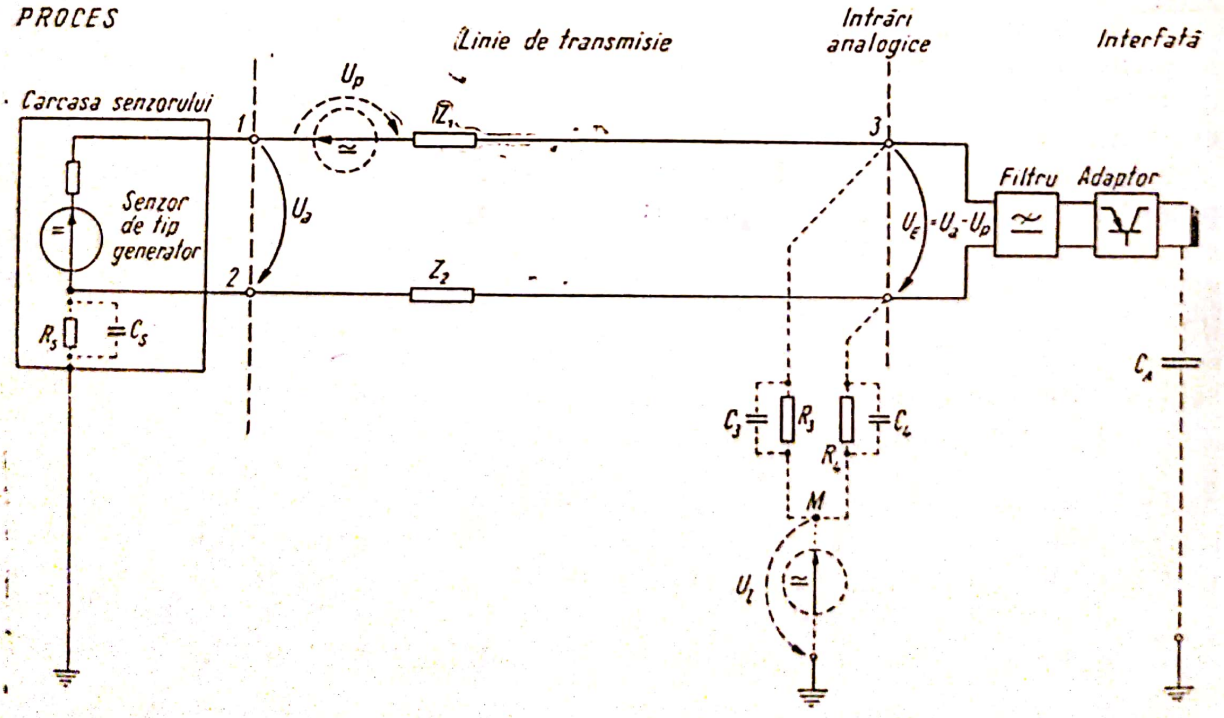


Fig. 11.21.

Scanned with OKEN Scanner

11.21). Datorită simetriei electrice ale celor două contacte ale intrării analogice, precum şi datorită capacităţilor mici faţă de pămînt ale întregului circuit (C_s, C_3 şi C_4) şi a rezistenţelor mari de izolaţie (R_s, R_3 şi R_4), efectul perturbator al tensiunii parazite asimetrice de linie U_l este neglijabil pînă la frecvenţe de cîţiva kilohertz. La frecvenţa cea mai întîlnită de 50 Hz (frecvenţa reţelelor locale de alimentare cu energie electrică), tensiunea parazită U_l are o amortizare în raport cu tensiunea de măsurat U_E mai mare decît 120 dB; deci efectul ei perturbator este practic neglijabil.

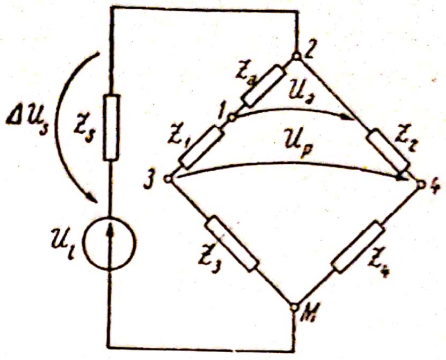


Fig. 11.22.

Această situaţie nu este însă întotdeauna prezentă. Astfel, dacă — prin defecte de izolaţie sau nesimetrii ale liniei de transmisie — apar diferenţe sensibile fie între impedanţele proprii ale liniei ($Z_1 \neq Z_2$) fie între impedanţele de izolaţie ale liniilor ($Z_3 \neq Z_4$, unde $Z_3 = R_3 + 1/j\,\omega C_3$ şi $Z_4 = R_4 + 1/j\,\omega\, C_4$), tensiunea asimetrică de linie se poate transforma într-o tensiune perturbatoare U_p ce se suprapune direct peste tensiunea utilă U_a produsă de senzor. Într-adevăr, dacă ne referim la schema echivalentă reprezentată în figura 11.22 a reţelei de transmisie (senzor-linie) din figura 11.21, rezultă că între punctele intrării analogice *3* şi *4* apare o tensiune perturbatoare U_p ce reiese din relaţia:

$$\mathcal{U}_p = (\mathcal{U}_l - \Delta\mathcal{U}_s) \cdot \left(\frac{\mathbb{Z}_1 + \mathbb{Z}_a}{\mathbb{Z}_1 + \mathbb{Z}_a + \mathbb{Z}_3} - \frac{\mathbb{Z}_2}{\mathbb{Z}_2 + \mathbb{Z}_4}\right),$$

care se obţine aplicînd schemei din figura 11.22 teoremele circuitelor electrice în cazul în care mărimile electrice au atît componente de curent continuu cît şi componente de curent alternativ. De aceea $\mathbb{Z}_1$, $\mathbb{Z}_2$, $\mathbb{Z}_3$, $\mathbb{Z}_4$, $\mathbb{Z}_s$ şi $\mathbb{Z}_a$ sînt impendanţe operaţionale (de forma: $\mathbb{Z} = R + sL + \frac{1}{sC}$), iar $\mathcal{U}_a$, $\mathcal{U}_p$, $\mathcal{U}_l$, şi $\Delta\mathcal{U}_s$ sînt transformatele Laplace ale tensiunilor U_a, U_p, U_l şi — respectiv — ΔU_s, unde ΔU_s reprezintă căderea de tensiune pe rezistenţa de izolaţie a senzorului faţă de carcasa sa (legată la masă). Prin aducerea la acelaşi numitor a fracţiilor dintre paranteze, relaţia de mai sus se scrie sub forma:

$$\mathcal{U}_p = (\mathcal{U}_l - \Delta\mathcal{U}_s)\,\frac{\mathbb{Z}_1\mathbb{Z}_4 - \mathbb{Z}_2\mathbb{Z}_3 + \mathbb{Z}_a\mathbb{Z}_4}{(\mathbb{Z}_1 + \mathbb{Z}_a + \mathbb{Z}_3)\cdot(\mathbb{Z}_2 + \mathbb{Z}_4)}.$$

De aici se deduce că dacă: $\mathbb{Z}_1\,\mathbb{Z}_4 = \mathbb{Z}_2\,\mathbb{Z}_3$ (ceea ce reprezintă condiţia de simetrie electrică, adică linii de transmisie identice $\mathbb{Z}_1 = \mathbb{Z}_2$ şi perfect izolate $\mathbb{Z}_3 = \mathbb{Z}_4$) şi impedanţa proprie a senzorului este neglijabilă ($\mathbb{Z}_a \to 0$), atunci nu apare o tensiune parazită $\mathcal{U}_p$ la bornele *3—4* ale intrării analogice. În caz contrar, adică de nesimetrie electrică a liniei de transmisie, paraziţii de linie U_l (v. figurile 11.21 şi 11.22) pot produce o perturbaţie la intrarea analogică (bornele *3—4*) cu atît mai mare cu cît $\Delta\mathcal{U}_s$ este mai mică, deci cu cît izolaţia senzorului faţă de carcasă (masă) este mai slabă.

Ex. 11—7. Un caz interesant îl prezintă senzorii pentru prelevarea temperaturilor dintr-un proces prin *termocupluri*. La un element termo-

Scanned with OKEN Scanner

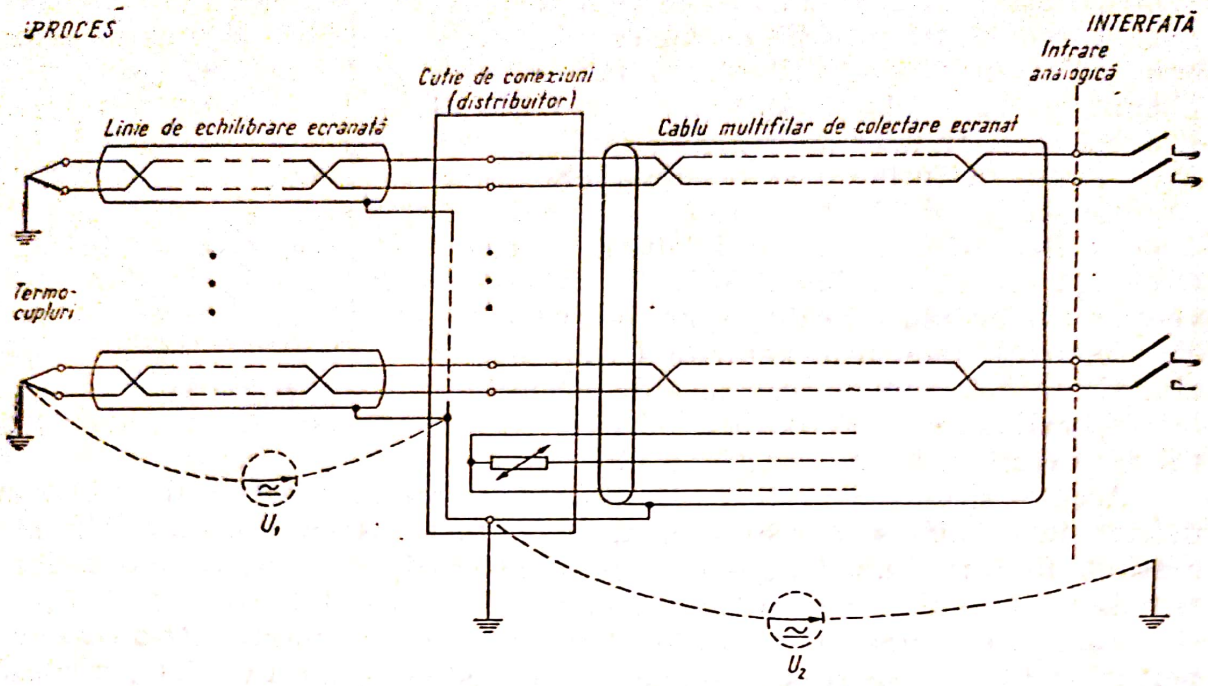


Fig. 11.23.

electric, linia de transmisie proces-interfață este asimetrică din două motive prezente simultan : *1)* existența liniilor de echilibrare ; *2)* legarea la pămînt a termocuplului, în locul de montare, printr-un conductor cu rezistență foarte mică (fig. 11.23). De aceea, în cazul termocuplurilor, tensiunilor simetrice de linie au totdeauna un efect perturbator direct, efectiv sesizabil la bornele intrărilor analogice ale interfeței.

Pentru a reduce cît mai mult efectul perturbator al paraziților de linie (simetrici), în special prin curenții capacitivi (de deplasare) provocați de diferența de potențial U_1 (v. fig. 11.23) între conexiunile la masă ale termocuplului și cablului de transmisie și U_2 — dintre masa cablului de transmisie și carcasa interfeței se iau următoarele măsuri speciale :

— pentru fiecare termocuplu, linia bifilară de transmisie este răsucită, ecranată static, cu ecranul și termocuplul legate la pămînt într-un punct comun ;

— dacă există mai multe termocupluri, liniile de echilibrare, bifilare și răsucite sînt ecranate separat pentru fiecare termocuplu în parte și conectate la o cutie cu borne plasată în apropierea instalației procesului. De aici, transmisia la interfața industrială se face printr-un singur cablu multifilar, cu perechi de fire răsucite, cîte una pentru fiecare termocuplu. Acest cablu de colectare are un ecran (blindaj) comun, legat la masă în același punct cu ecranele liniilor de echilibrare (v. fig. 11. 23) ;

În succesiunea diverselor termocupluri, bornele de legare din cutia de conexiuni (distribuitor) trebuie alese cu grijă, nu oricum ci în acea ordine pentru care, în cazul unor distribuții de temperatură nefavorabile (diferite), să nu rezulte componente perturbatoare din tensiunile termoelectromotoare ce apar la bornele de conectare.

Transmisia semnalelor analogice. În cazul procedeului de transmisie, proces-interfață, a semnalelor analogice prin perechi de conductoare și

Scanned with OKEN Scanner

fără modulația unei purtătoare, un rol deosebit îl au numai doi factori: adaptarea impedanței liniei și suprimarea zgomotelor.

Adaptarea impedanței liniei de transmisie. Metoda cea mai indicată pentru asigurarea unei bune adaptări constă în a utiliza senzori care să aibă, ca semnal de ieșire, curentul electric (nu tensiunea). În acest caz, semnalul emis — un curent electric — este transmis prin conductorul liniei pînă la intrarea analogică a interfeței, unde este montat un rezistor potențiometric pentru transformarea semnalului de curent într-o tensiune electrică. În acest fel, efectul impedanței liniei în raport cu impedanța potențiometrului de la intrarea interfeței este neglijabil, ceea ce face ca semnalul emis de senzor (un curent electric) să nu depindă decît de mărimea de măsurat și nu de linia de transmisie.

În cazul unui semnal sub formă de tensiune electrică, mai ales dacă nu are o valoare de repaus mai mare și dacă linia de transmisie este lungă, efectul căderilor de tensiune prin transmisie directă poate duce la erori de etalonare. În plus, în cazul semnalelor de tensiune, impedanța senzorului și a liniei sînt în serie, nu numai între ele, ci și cu impedanța de intrare a interfeței, ceea ce micșorează supleţea adaptării.

Evident, rezistența maximă admisibilă a liniei proces-interfață depinde de caracteristicile sistemului de intrare/ieșire al calculatorului. În prezent, la majoritatea calculatoarelor, impedanța proprie a liniei de transmisie trebuie să fie în jurul a 250 Ω (cu limite extreme, după tip de calculator, de la 150 Ω la 1 000 Ω).

Suprimarea zgomotelor. Mai bine spus, este vorba de reducerea efectelor zgomotelor sub limita clasei de precizie a senzorului utilizat. Din aliniatul precedent, am desprins ideea că suprimarea zgomotului nu trebuie localizată numai la senzor sau numai la linia de transmisie; ea trebuie făcută pe întreg sistemul de intrare/ieșire, în corelare și cu mediul ambiant, instalația procesului etc.

Așa cum am mai arătat, atenția cea mai mare trebuie acordată diminuării paraziților asimetrici, din interiorul rețelei de intrare, ceea ce — în general — se realizează prin eliminarea cuplajelor de orice fel, de conducție, capacitive și inductive, prin: izolarea perfectă, contacte inoxidabile, ecranare, răsucirea perechilor de fire, raportarea la un potențial comun, filtrare specială la intrarea interfeței. Ecranarea magnetică se face cu table speciale de oțel feromagnetic și se aplică numai surselor de cîmp magnetic (transformarea de semnal.) Ecranarea electrostatică se face, eficient, fie cu tresă (împletitură) de cupru sau de oțel, fie cu bandă de plastic metalizată.

Suprimarea paraziților simetrici dintre linie și pămînt se face prin: legarea la masă (pămînt) într-un singur punct; izolări față de masă cît mai perfecte ale liniei și întregului aparataj aferent; ecranarea și blindarea liniei și aparatelor; eliminarea curenților de conturare, dispersie etc. (prin umezeală, praf, ionizări ale mediului etc.).

În afara celor de mai sus, se impune acordarea unei atenții speciale și locului de montare a aparatelor sau de „pozare" a liniilor de transmisie proces-calculator. Astfel, amplasarea aparatelor și liniilor de transmisie trebuie făcută cît mai departe de cîmpurile electromagnetice perturbatoare cum sînt, de exemplu, cele din jurul motoarelor electrice dar și termice (care lucrează cu scîntei la bujii, aprinderi periodice etc.), transformatoarele

Scanned with OKEN Scanner

electrice, rețelele electrice ș. a. m. d. În sfîrșit, la lucrările de montaj se va avea grijă ca în cablurile multifilare colectoare, cu ecran comun, să nu se introducă în afară de semnale analogice și fire cu semnale numerice.

Transmisia semnalelor numerice. Toate sursele de perturbații, prezentate anterior, au efecte similare și în cazul transmisiilor numerice. Totuși, în cazul semnalelor numerice există mai multe posibilități de discriminare a paraziților de semnale utile, atît prin construcția și montajul rețelei de transmisie, cît și prin filtrări, mod de tratare (operare) a intrărilor numerice prin programele de control direct și chiar prin programele de prelucrare.

Astfel, în cazul senzorilor de tip contact, care emit semnale sub formă de treaptă, eliminarea perturbațiilor se poate face cel mai simplu prin utilizarea unei trepte cu nivel mare, adică un „salt" mare de tensiune ca, de exemplu, de la 50 V pentru starea „1" (DA) la O V pentru starea „O" (NU); la un astfel de salt, orice tensiune parazită — care nu poate ajunge la astfel de valori mari — este cu ușurință eliminată de filtrele de la intrarea interfeței.

În cazul semnalelor sub forma unor trenuri de pulsuri, lucrurile se prezintă diferit după cum este vorba de: *1)* numărarea pulsurilor (ca „obiecte"), *2)* măsurarea frecvenței pulsurilor și *3)* decodificarea semnalelor codificate în binar. În ultimul caz, paraziții pot fi ușor detectați prin pulsurile părții de control — validare; în cazul *2)* preluarea semnalului corect se face adesea prin repetarea operației de măsurare a frecvenței. În primul caz, operația este aparent mai dificilă (numai dacă nu intervin paraziți sistematici); în general, eliminarea efectului perturbator se face prin analiza intervalului dintre pulsuri, a amplitudinii și formei pulsului (prin filtrele de la intrare).

Executori — introducerea comenzilor

În cadrul acestui paragraf ne vom referi — în măsura necesară unui utilizator — la modul cum sînt valorificate direct rezultatele determinate de calculator prin prelucrarea valorilor cu privire la proces, prelevate prin senzori. Așa cum s-a arătat în subcapitolul 10.4, există mai multe modalități de valorificare a acestor date: listări și afișări prin imprimatele și „display"-urile sistemelor standard de intrare/ieșire, destinate operatorului uman; exploatarea și ținerea la zi a unor fișiere, baze sau chiar bănci de date în vederea regăsirii operative a informației; avertizări, semnalizări etc. direct la proces și „lansarea" unor semnale de comandă pentru aparatura de automatizare din proces (elemente de acționare/execuție și elemente de reglare convenționale, pe care ne-am permis să le denumim generic prin numele de *executori*), în vederea conducerii în timp real a proceselor. În cele ce vor urma, vom avea în vedere numai această modalitate, insistînd asupra procedeului de „introducere în proces" a comenzilor elaborate de calculator sub forma unor semnale analogice sau/și numerice elaborate la „ieșirile" interfeței industriale, proces-calculator.

Executori analogici. La ieșirile analogice ale interfeței industriale, comenzile pot fi sub forma unor semnale electrice analogice de tensiune sau, cel mai adesea, de curent.

În cazul comenzilor sub formă de tensiuni electrice, efectul liniei de transmisie directă interfață-proces (mai ales atunci cînd distanțele sînt

Scanned with OKEN Scanner

mai mari decît 100 ÷ 200 m) este însemnat prin căderea de tensiune ce o produce. În astfel de situaţii, adaptarea dintre impedanţele : de ieşire din interfaţă, ale liniei de transmisie şi ale executorilor analogici este mai dificilă de realizat, uneori plasîndu-se la intrarea în executor a unui amplificator de tensiune (v. *A* în fig. 11.3), cu bandă foarte largă de trecere, ceea ce complică întrucîtva instalaţia şi măreşte posibilităţile apariţiei unor perturbaţii.

De aceea, se preferă ieşirile analogice sub formă de curent electric — cu un domeniu de lucru de la 0 la 20 mA — pentru care problemele de adaptare ale impedanţei liniei de transmisie directă nu sînt dificile nici la distanţe de 500 ÷ 700 m.

Executorii analogici, din proces, sînt aproape în exclusivitate aparate electrice de măsurat, fie indicatoare (cu ac sau cu spot luminos), fie cu înregistrare grafică în funcţie de timp. Este vorba, deci, de aparate clasice (convenţionale) de măsurat tensiuni sau curenţi electrici, însă etalonate corespunzător mărimii măsurate [10]. Singura observaţie — însă foarte importantă — ce trebuie avută în vedere de utilizator este aceea că se vor folosi astfel de aparate de măsurat prin intermediul calculatorului numai pentru acei parametri ai procesului care nu pot fi măsuraţi direct, local (în proces), prin traductor-aparat de măsurat, fie pentru că nu există un traductor corespunzător, fie pentru că valoarea măsurată şi indicată în proces trebuie determinată prin calcule speciale în funcţie de valorile altor parametri tehnologici prelevate din proces. Exemple de astfel de situaţii sînt : dacă, pentru introducerea în calculator a mai multor valori, se foloseşte o claviatură conectată la intrarea digitală a interfeţei, atunci dintr-o serie de mărimi poate fi aleasă numai una (în funcţie de anumite criterii), valorile ei fiind transmise aparatului de indicat printr-o ieşire analogică ; la înregistratoarele pentru mai multe curbe există posibilitatea de a indica, printr-un singur aparat, valorile pentru toate curbele trasate în paralel, obţinute prin semnale de la o singură ieşire analogică a interfeţei (redarea sincronă a măsurătorilor de către calculatorul de proces este comandată de un comutator al aparatului de la locul de măsurat — prin multiplexarea ieşirii analogice [24]).

Mai trebuie avut în vedere faptul, de altfel evident, că în cazul scoaterii din funcţiune a calculatorului, aparatele de măsurat conectate la ieşirile analogice ale interfeţei nu vor mai indica nici ele.

Executori numerici (digitali). Semnalele numerice produse de calculator la ieşirile numerice ale interfeţei sînt folosite în proces pentru comanda unor executori numerici ca : relee (de obicei binare), contactoare, cuplaje magnetice, servomotoare electrice de tip pas cu pas, lămpi de semnalizare şi alte asemenea elemente de reglare şi avertizare (indicare) ce lucrează binar (v. fig. 11.24).

În funcţie de necesităţile referitoare la puterea de comandă, frecvenţa de conectare, nivelul de tensiune electrică etc., semnalul de comandă de la eşirile numerice ale interfeţei provin fie de la bornele cu tensiune continuă ale unor circuite logice cu tranzistoare, fie de la reţele cu rezistoare alimentate de la un registru numeric paralel de memorie, fie de la contactele unor relee (se utilizează de obicei contacte cu mercur).

Scanned with OKEN Scanner

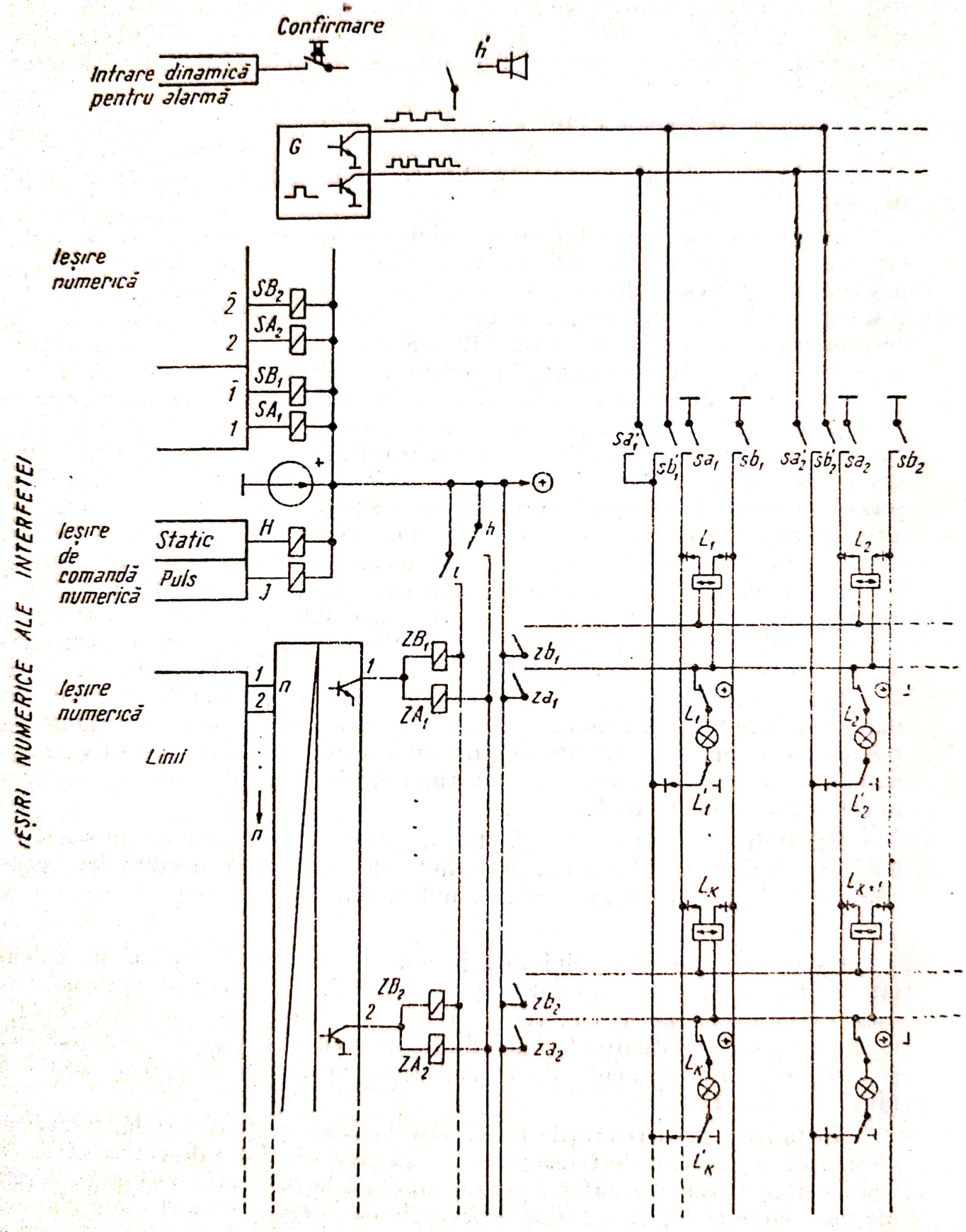

IEȘIRI NUMERICE ALE INTERFEȚEI
Confirmare
Intrare dinamică pentru alarmă
h'
G
Ieșire numerică
2̂
SB2
2
SA2
1̂
SB1
1
SA1
+
Ieșire de comandă numerică
Static
H
Puls
J
h
t
Ieșire numerică
1
2
n
Linii
ZB1
ZA1
ZB2
ZA2
zb1
za1
zb2
za2
sa'1
sb'1
sa1
sb1
sa'2
sb'2
sa2
sb2
L1
L2
LK
LK+1
L'1
L'2
L'K

Fig. 11.24.

În ceea ce privește forma semnalului de la ieșirile numerice, aceasta poate fi: o treaptă („salt") de tensiune electrică (comandă executori ca: bobinele releelor, elemente de acționare electromagnetice, cuplaje magnetice etc.); un singur puls de tensiune (comandă executori ca: relee cu acțiune repetată, contactoarele de pornire a motoarelor sau servomotoarelor etc.); un tren de pulsuri de tensiune cu succesiune în timp (comandă executorii de poziționare cum ar fi, de exemplu, un servomotor electric pas cu pas) și n pulsuri paralele codificate — cu 2^n valori distincte — care pot comanda executori formați din convertoare de cod sau matrice cu relee și contacte (v. *Ex. 11—8*).

Executorii care participă efectiv la modificarea procesului și sînt comandați de semnalele provenite de la ieșirile numerice ale interfețelor trebuie să fie astfel concepuți încît atunci cînd calculatorul iese din funcțiune (și deci nu mai furnizează semnale de comandă) ei să treacă pe poziții „sigure", care să evite crearea unor situații periculoase pentru proces. La readucerea calculatorului în stare de funcționare, ieșirile numerice rămîn în poziția de repaus. După aceasta, la ieșirile numerice ale interfeței pot fi produse semnale prin acțiunea programului de comandă directă.

Ex. 11—8. În încheiere, prezentăm ca exemplu un executor tipic pentru conducerea cu calculatoare a proceselor și anume, o *matrice de relee* (fig. 11.24). Aceasta este utilizată atunci cînd printr-o ieșire numerică a interfeței trebuie comandate un mare număr de elemente de acționare și/sau reglare binare (contactoare, supape, robinete cu caracteristici de tip releu: „închis" sau „deschis", lămpi de semnalizare, avertizoare acustice etc.). Prin folosirea matricelor de relee se reduce mult numărul ieșirilor numerice ale interfeței în comparație cu transmisia directă individuală (pentru fiecare executor separat).

Releele folosite în nodurile matricei, la intersecția liniilor de intrare cu coloanele de ieșire, sînt — cu unele rare excepții — relee polarizate, care își păstrează nemodificată starea de cuplare și după suprimarea tensiunii de comandă.

În figura 11.24 este prezentată schema de principiu — funcțională — a unei matrice cu relee pentru comanda unor lămpi de semnalizare, redusă numai la liniile și coloanele primei cifre — cea mai puțin semnificativă (lămpile L_1 și L_2) și ultimei cifre — adică cea mai semnificativă (lămpile L_K și L_{K-1}). Lămpile folosite sînt lămpi de semnalizare cu lumină intermitentă. După cum se constată din figura 11.24, fiecare lampă de semnalizare este cuplată cu ajutorul unui releu polarizat L_i, cu contactele L_i și L_i' ($i = 1, 2, \ldots, K, K+1$). Ambele înfășurări de comandă ale fiecărui releu polarizat sînt „plasate" într-un punct de încrucișare a liniilor și coloanelor. La rîndul lor, liniile și coloanele sînt comandate (puse sub tensiune) de contactele auxiliare za_j și zb_j ale bobinelor ZA_j și — respectiv — ZB_j (pentru linii) și sa_j și sb_j ale bobinelor SA_j și — respectiv — SB_j (pentru coloane), unde $j = 1, 2, \ldots$. Bobinele ZA și SA servesc la cuplarea lămpilor semnalizatoare, iar SB și ZB la decuplarea lor.

Conform diagramei din figura 11.25, în care sînt reprezentate formele de variație în timp ale diferitelor semnale numerice de comandă pentru schema din figura 11.24, modul de semnalizare se realizează prin atragerea de către semnalele de la ieșirile numerice aplicate liniilor a unui releu ZA și legarea prin releul H (v. fig. 11.24) a unei bobine SA. În acest fel, lampa

Scanned with OKEN Scanner

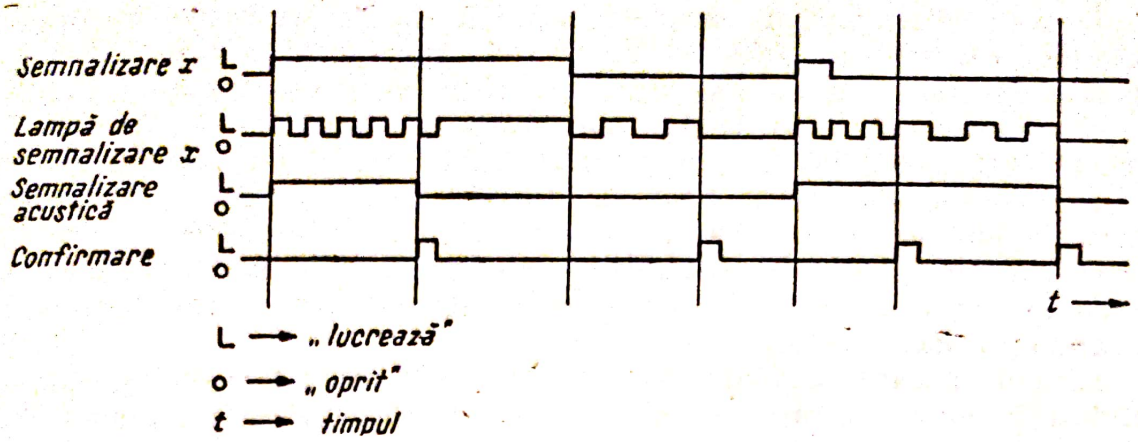


Fig. 11.25.

de semnalizare — care se află la „intersecția" liniilor cu coloanele — primește un semnal intermitent rapid produs de generatorul de pulsuri G (v. fig. 11.24).

La confirmarea unui semnal de alarmă, se produce cuplarea scurtă a bobinei J și decuplarea lui H. Atunci bobinele ZA se desfac și ZB sînt cuplate, ceea ce determină acționarea bobinelor L astfel încît lampa respectivă va lumina permanent (ceea ce reprezintă o indicație de „atenție").

11.3. SISTEMUL DE INTRARE/IEȘIRE

Calculatoarele de proces prezintă particularități cu totul specifice în ceea ce privește sistemul lor de intrare/ieșire, legat de interfața unitate centrală de prelucrare-proces, care asigură conectarea la unitatea centrală a unei largi game de dispozitive periferice necesare conducerii unui proces industrial sau de laborator, oricare ar fi acesta.

În general, există *patru nivele* ale interfeței unitate centrală-proces, așa cum se arată schematic în figura 11.26.

Nivelul 1 se referă la *interfața unității centrale de prelucrare* (procesorul), care asigură schimburile de informație între organele periferice exterioare și procesor prin intermediul registrului acumulator, registrului de acces-memorie și al sistemului de întreruperi prioritar. La acest nivel, utilizatorul dispune de cel mai mare „grad de libertate" pentru „conceperea" sistemului de intrare/ieșire.

Nivelul 2 se referă la *interfața prin legătură directă* a sistemului de intrare/ieșire, care asigură „administrarea" schimburilor de informație între unitatea centrală și elementele de cuplare la dispozitivele periferice. La acest nivel, modalitățile schimbului de informație, adică „maniera" utilizării interfeței la nivelul 1 (precedent), sînt bine definite. Ca urmare, la nivelul 2 utilizatorul dispune de cel mai restrîns grad de libertate pentru adaptarea dispozitivelor periferice de proces.

Nivelul 3 se referă la *interfața comună pentru conexiunile subansamblurilor specializate*, care permite controlul și comanda dispozitivelor spe-

Scanned with OKEN Scanner

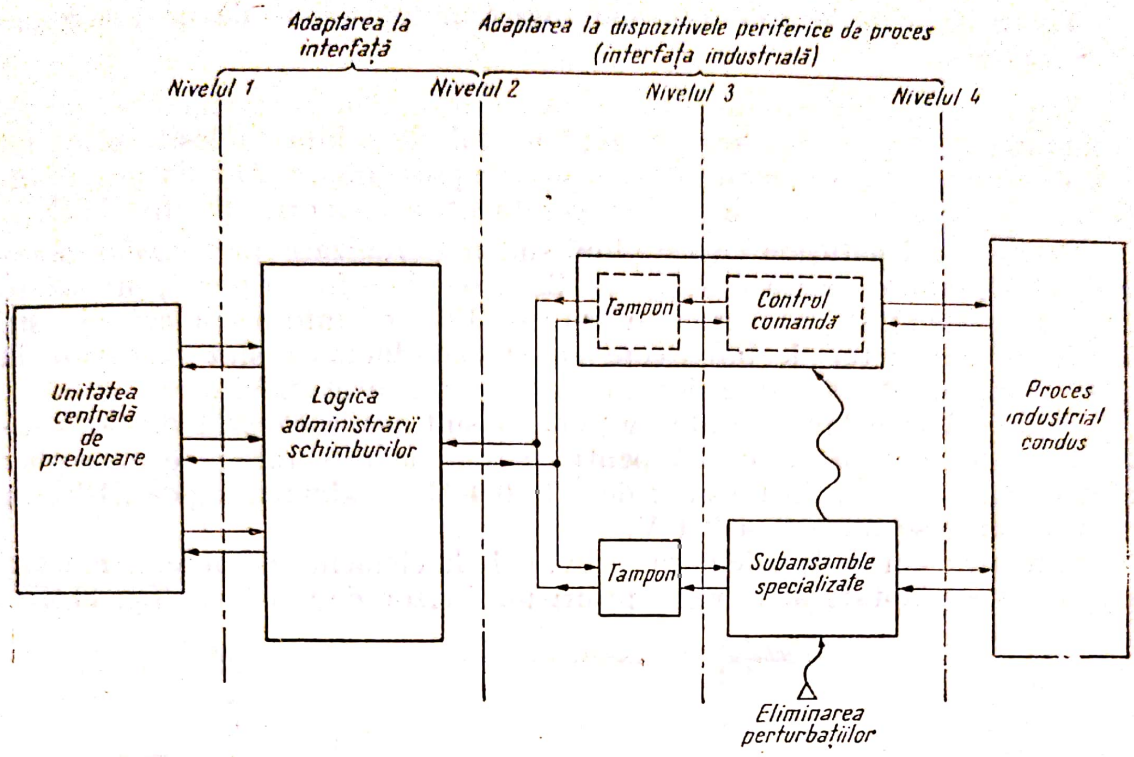


Fig. 11.26.

cializate (cu tehnologie electronică complexă) cum sînt, de exemplu, legăturile de măsură sau ieşirile analogice. Bineînţeles, această interfaţă foloseşte modalităţile de schimb definite de către interfaţă la nivelul 2.

Nivelul 4 se referă la *interfaţa specială pentru terminalele din proces*, care asigură conexiunea cu instrumentaţia de proces considerată ca dispozitive periferice speciale. La nivelul 4 al interfeţei industriale se întîlnesc semnalele prelevate din proces sau dirijate către proces (cele ce au fost prezentate în subcapitolul 11.2). După cum am şi văzut, la acest ultim nivel utilizatorul nu are — practic — nici o libertate de adaptare, deoarece informaţia „schimbată" (calculator ⇄ proces) este strict determinată de specificul dispozitivelor periferice ce trebuie utilizate în proces. La acest nivel, interfaţa este „materializată" prin liniile de transmisie şi bornele de conexiuni.

În legătură cu aceasta, putem preciza că interfaţa la nivelul 1 este materializată prin cabluri de legătură, interfaţa la nivelul 2 — prin conectori de legătură la o „linie de vehiculare" (linie „omnibuz") care constau din module (plăci) cu circuite integrate cu elemente active din semiconductori, interfaţa la nivelul 3 — prin cabluri de conexiuni locale (unde nu există puncte de „disponibilităţi" ale informaţiei). Din cele de mai sus, comparînd între ele şi figurile 11. 1 cu 11.26, rezultă că între nivelele 2 şi 4 se găseşte materializată (prin echipamente) ceea ce noi am numit interfaţa industrială, adică interfaţa calculator-proces, cu „intrările" şi „ieşirile" pentru proces (nivelul 4) şi „legăturile" cu procesorul (nivelul 2).

De aceea, în cadrul acestui subcapitol ne vom referi — în maniera necesară utilizatorilor — la tipul de schimburi posibile între procesor şi exterior (la nivelul 1 — nivelul 2) şi la interfaţa industrială (nivelul 2 — nivelul 4).

Scanned with OKEN Scanner

Tipuri de schimburi de informaţie între unitatea centrală de prelucrare şi exterior

Schimbul de informaţie între procesor (unitatea de prelucrare + memoria internă) şi periferice se face, după modul de schimb folosit, prin trei linii diferite: *1)* linia pentru intrările/ieşirile programate, *2)* linia pentru întreruperile prioritare şi *3)* linia de acces direct la memorie (v. fig. 11.28).

Realizarea legăturilor. Aceste linii sînt materializate prin cabluri standard (ISO) cu o capacitate de 44 sau 25 perechi de fire răsucite, prevăzute la fiecare extremitate cu conectoare cu fişe. Fiecărei informaţii logice îi este asociat un conductor de întoarcere (aceste conductoare sînt regrupate la fiecare extremitate a cablurilor în interiorul conectorului).

Valorile logice binare (informaţionale) sînt realizate prin nivele standardizate de tensiune electrică pentru circuitele integrate; de exemplu: valoarea logică „0" prin tensiuni de 0 la 0,4 V şi valoarea logică „1" prin tensiuni cuprinse între 5 şi 2,4 V.

Semnalele emise de calculator provin de la elemente de putere. Fiecare legătură este adaptată la recepţie printr-un divizor de tensiune (fig. 11.27).

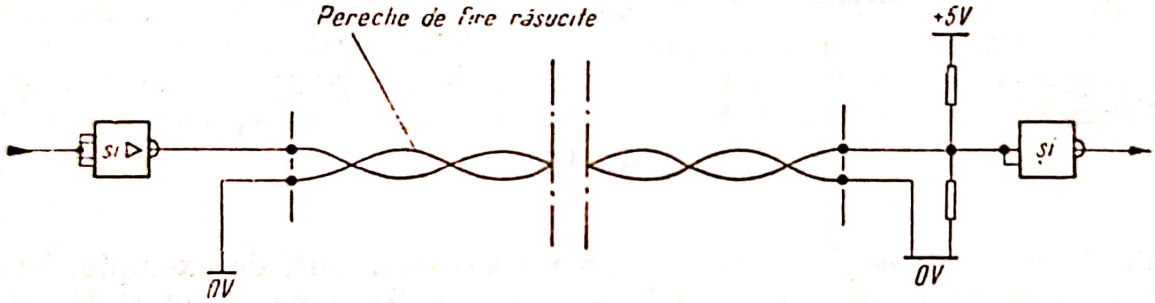


Fig. 11.27.

Schema bloc generală a interfeţei unui procesor, la nivelul 1, este reprezentată în figura 11.28 în care: A reprezintă un registru acumulator de intrare/ieşire; C — registru de instrucţiuni; M — registru memorie şi ADM_{ij} — prize pentru acces direct la memorie.

Într-o formă mai generală, toate semnalele sînt transmise astfel încît în repaus să reprezinte nivelul logic „1"; aceasta din motive tehnologice. Ele sînt active, deci, la nivelul logic „0" (prin urmare sînt transmise sub formă complementară $\overline{0} = 1$).

În cele ce vor urma, vom nota cu E semnalele care ies din calculator şi cu I cele care intră.

Linia de intrare/ieşire programată. Această linie este formată din cabluri standard legate la prizele de intrare/ieşire programată (de exemplu, pentru cazul din figura 11.28 sînt necesare două cabluri cu cîte 44 perechi de fire răsucite pe fiecare cablu, conectate la prizele I/E_1 şi I/E_2).

Prin această linie se transmit toate semnalele ce intervin în cursul schimburilor controlate de instrucţiunile de intrare/ieşire din acumulatorul A, accesibile prin programe: EA — ieşirea conţinutului acumulatorului şi IA — intrarea informaţiei în acumulator.

Structura informaţiei schimbate constă din trei părţi:

— semnalele informaţionale furnizate şi/sau luate de/din calculator prin acumulatorul A potrivit sensului de schimb (IA sau EA). Această parte

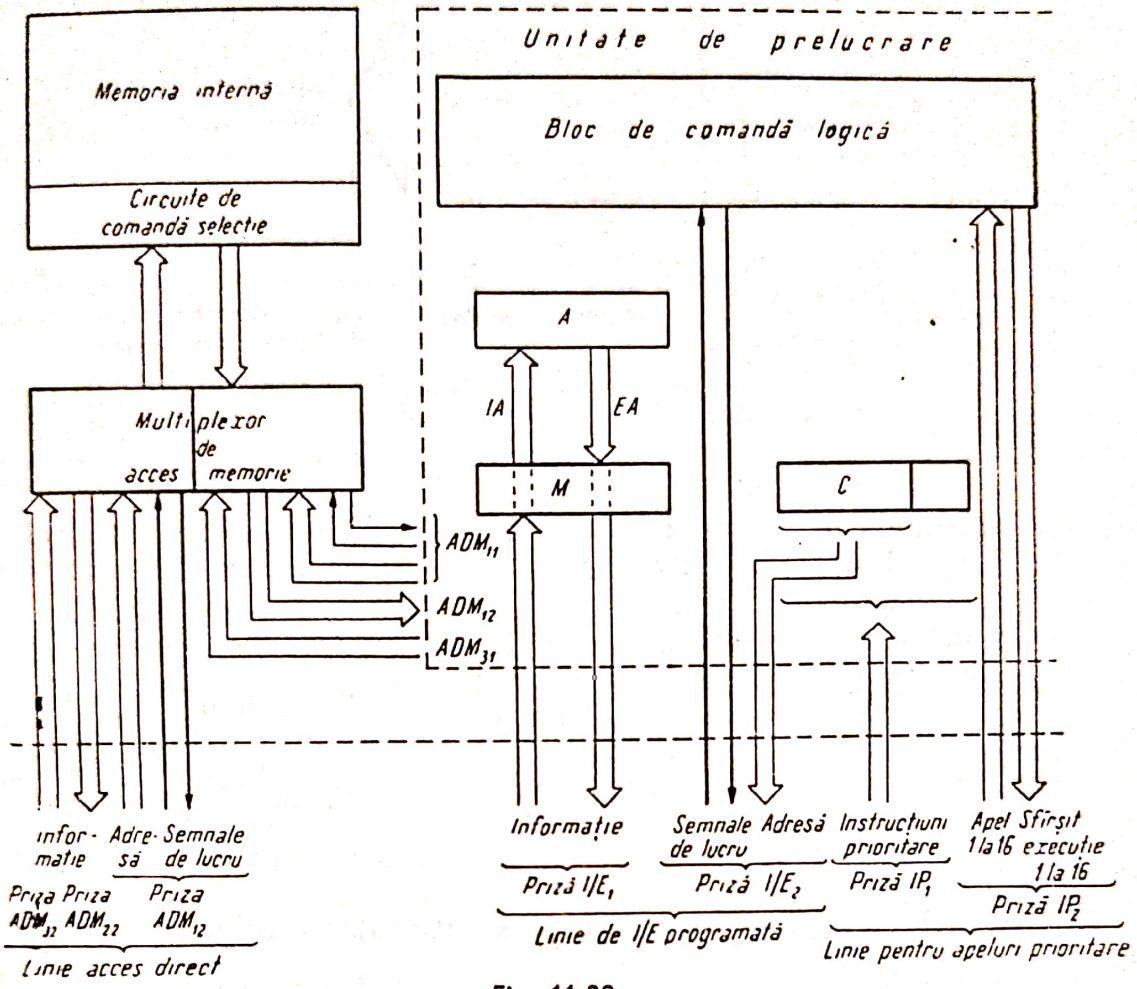


Fig. 11.28.

poate avea 18 bits utili (informaţionali) şi 1 bit pentru controlul parităţii în sensul ieşirii ;

— semnalele de identificare a adresei perifericului avut în vedere (partea operantului din registrul C — v. fig. 11.28), care poate fi format din 12 bits;

— semnalele de lucru necesare pentru definirea stării perifericului sau a validităţii codului primit şi a validităţii transferului însăşi.

Revenind la cazul din figura 11.28, constatăm că cele două cabluri ale liniei de I/E programată trebuie repartizate astfel : un cablu numai pentru transferul informaţiei, altul pentru adrese şi semnalele de lucru, aşa cum se arată mai detaliat în figura 11.29, în care : M_0 la M_{19} reprezintă celulele în registrul memoriei; C_0 la C_{11} — celule în registrul C şi IX_0 la IX_{18} — bits informaţionali de ieşire.

Linia pentru întreruperi prioritare. Această linie este formată tot din cabluri standard conectate la prizele de întreruperi (IP_1 şi IP_2 în exemplul din figura 11.28, unde pentru această linie sînt necesare două cabluri, fiecare cu cîte 44 perechi de fire).

Prin această linie sînt transmise semnalele produse în timpul schimbului de control pe baza instrucţiunilor prioritare, inaccesibile prin program, şi care sînt, de exemplu : UR — „ruptură urgentă", NP — „numărătoare prioritară", EM — „ieşire din registrul memorie" şi IM — „intrare în

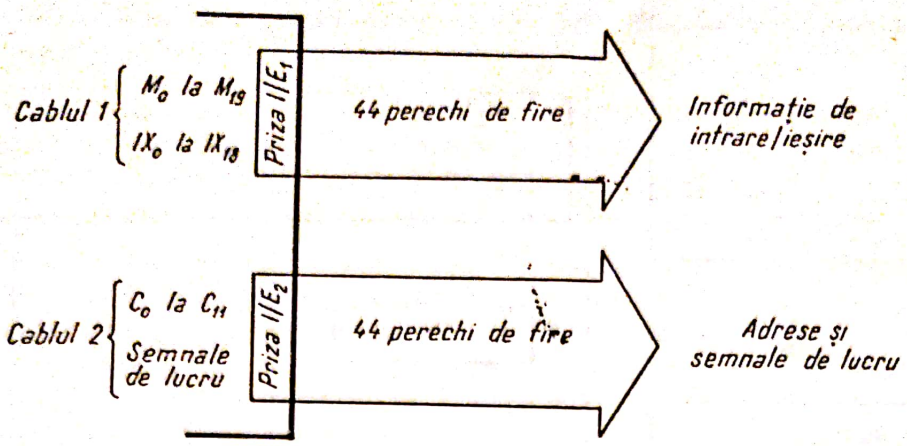


Fig. 11.29.

registrul memorie". Aceste instrucțiuni sînt introduse în registrul C (v. fig. 11.28) al unității centrale prin sistemul de întreruperi prioritare.

Semnalele schimbate prin această linie pot fi de cinci tipuri:

— semnale de apel ale nivelului de întrerupere (notate cu AP_i în figura 11.30) care pot avea pînă la 16 bits;

— semnale de selectare a nivelului de întrerupere (SP_i — fig. 11.30) care pot avea pînă la 16 bits;

— semnale de codare a instrucțiunilor prioritare (CIP_i — fig. 11.30) formate din: 12 bits pentru adresă, 2 bits pentru modul de adresare, 2 bits pentru codul instrucțiunii și 3 bits pentru extinderea codului (în total 19 bits);

— semnale de sfîrșit de execuție a întreruperii prioritare (EP_i — fig. 11.30) care pot avea 16 bits;

— semnale diverse necesare pentru comanda și inițializarea sistemului de întreruperi prioritare.

Considerînd exemplul din figura 11.28, se vede că pentru această linie trebuie două cabluri distincte: unul necesar ca suport al informației de codare a instrucțiunilor prioritare și de selectare a nivelului de întrerupere, celălalt pentru informația de apel și de sfîrșit de execuție a întreruperilor prioritare (fig. 11.30).

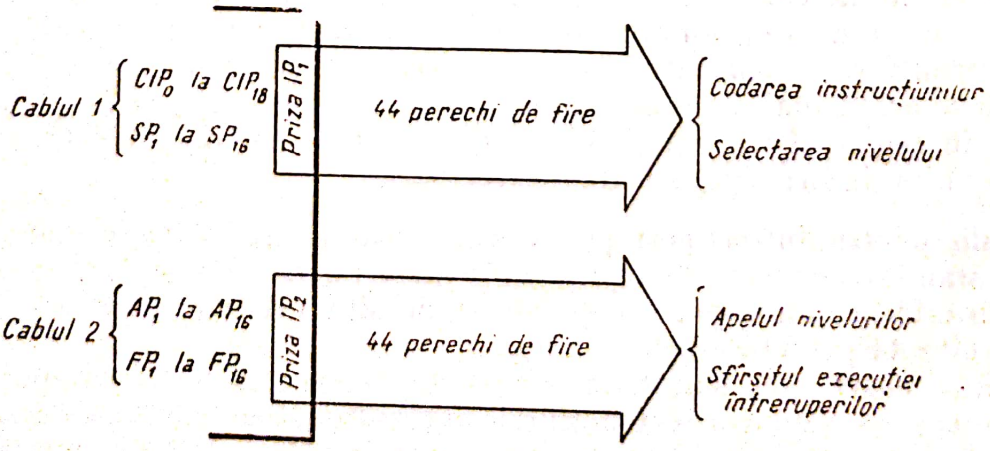


Fig. 11.30.

Scanned with OKEN Scanner

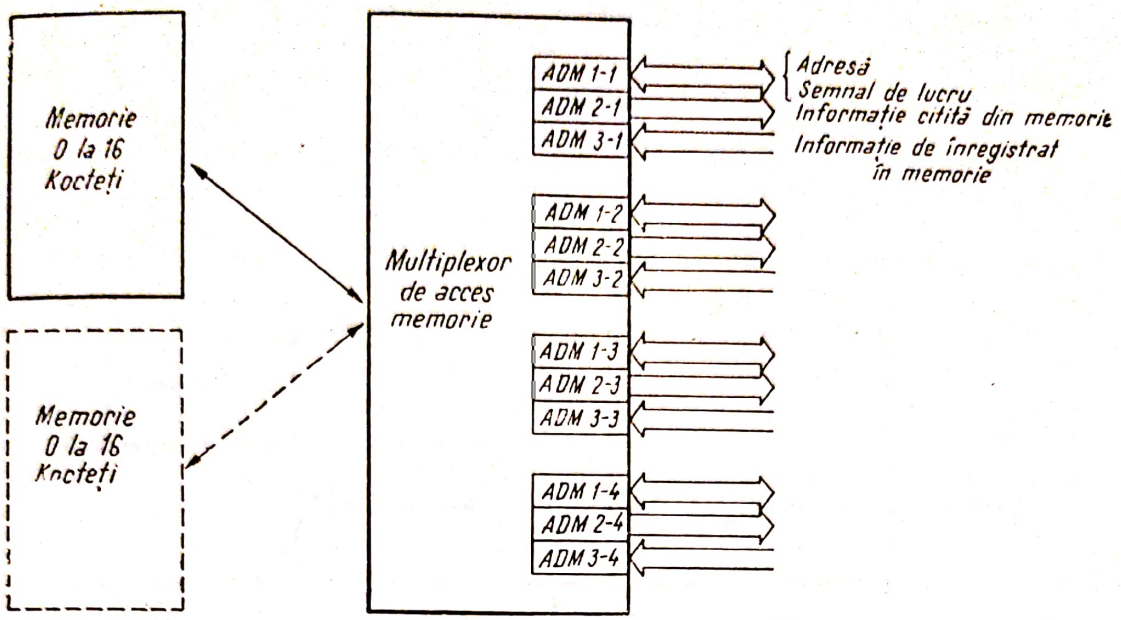


Fig. 11.31.

Linia de acces direct la memoria internă. Considerînd exemplul din figura 11.28, această linie este formată din trei cabluri standard, fiecare cu cîte 25 perechi de fire răsucite, legate la prizele ADM_1, ADM_2, ADM_3 ale dispozitivului multiplexor pentru accesul la memoria internă.

Multiplexorul pentru accesul la memorie este plasat înaintea memoriei propriu-zise și poate primi 2 la 4 linii de acces direct la memoria internă. El este format dintr-un selector care baleiază aceste linii de acces, oprindu-se la linia pentru care găsește o cerere în vederea „servirii" ei. În general, un astfel de selector poate deservi un bloc de memorie cu capacitatea de 16 Kocteți; dacă — în cazul unei memorii interne cu capacitatea mai mare decît 16 Kocteți — funcționează în paralel mai multe selectoare, atunci acestea pot lucra autonom și simultan.

Multiplexorul, legat pe de o parte la memorie, are, pe de altă parte, și prizele necesare pentru primirea a pînă la patru linii de acces direct la memorie. Pe aceste linii pot fi plasate, indiferent cum, una sau mai multe unități de prelucrare și unul sau mai multe canale de acces direct.

Prin liniile de acces direct la memorie sînt transmise (fig. 11.31):

— adresa memoriei la care s-a creat accesul (prin 15 bits);

— semnalele de lucru pentru cererea accesului la memorie sau, în sens contrar, de prevenire a transferului informației;

— informația citată din memorie (prin 19 bits informațional + 1 bit de control al parității);

— informația care trebuie înregistrată în memorie (19 + 1 bits).

Interfața industrială

Așa cum am mai arătat (v. fig. 11.1 și 11.26), calculatorul de proces poate fi privit ca o instalație complexă formată dintr-un *calculator automat* (pentru prelucrarea datelor), *dispozitive periferice de proces* (pentru control-comandă) și un sistem intermediar de conectare „on-line" a perifericelor la calculator, care poartă adesea numele generic de *interfață*.

Scanned with OKEN Scanner

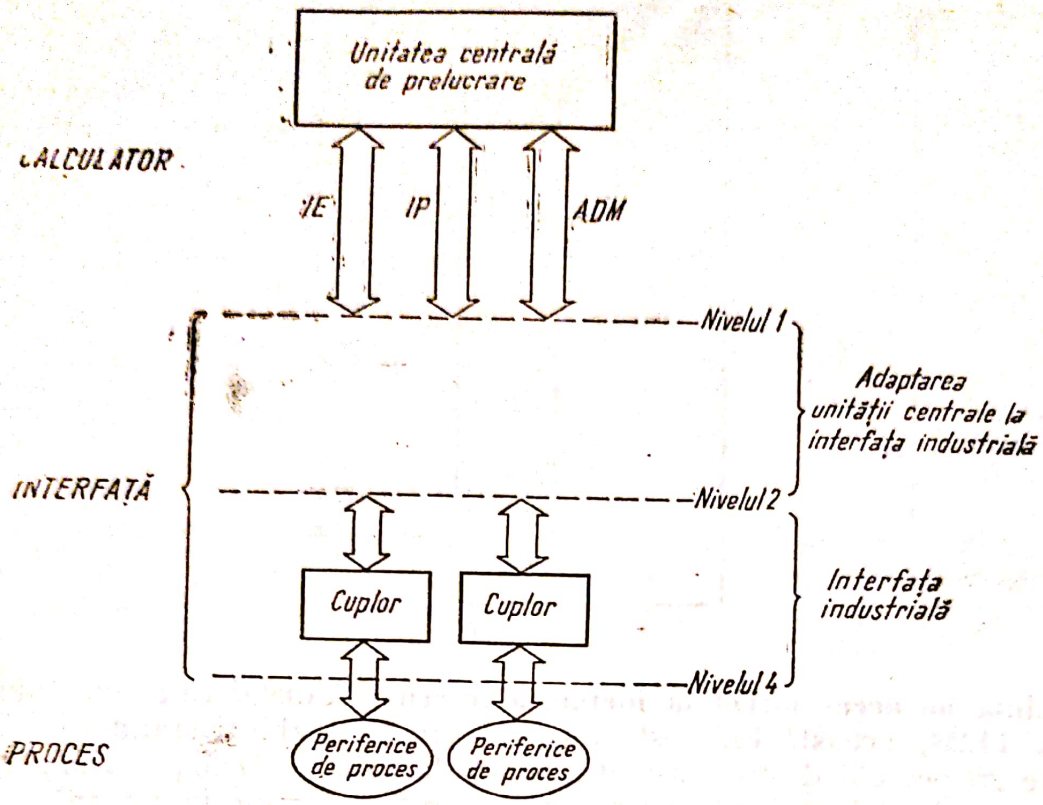


Fig. 11.32.

Interfața, datorită lanțului de adaptări succeisve pe care le realizează, are mai multe nivele (v. fig. 11.26). *Dispozitivele de adaptare la perifericele de proces poartă, în ansamblul lor, denumirea de interfață proces-calculator sau — mai scurt — interfața de proces sau* **interfața industrială.** Interfața industrială este formată dintr-o serie de elemente structurale de adaptare și transmisie a semnalelor care, grupate pe anumite categorii de periferice de proces (senzori analogici, senzori numerici, executori analogici, executori numerici etc.) formează dispozitive specifice de adaptare denumite adesea și **cuploare** (fig. 11. 32, în care: *IE* — linia de intrare/ieșire programată, *IP* — linia de apeluri prioritare prin întreruperi și *ADM* — linia de acces direct la memorie).

Generalități. În scopul fixării ideilor fundamentale, revenim mai întîi asupra unor noțiuni prezentate în cadrul acestui subcapitol.

La un calculator de proces, unitatea centrală de prelucrare și sistemul său de exploatare realizează schimburi de informație cu dispozitive periferice foarte diverse, care pot fi de două categorii distincte: *1)* periferice convenționale (cititoare de cartele, cititoare de bandă perforată, imprimante rapide, unități ale memoriei externe etc.) și *2)* periferice industriale, de proces (instrumentația de control — comandă: senzori și executori).

Informația emisă (sau primită) către (respectiv de la) aceste periferice este caracteristică fiecărui tip în parte, prin elemente ca: natura semnalului fizic purtător de informație (analogică sau numerică); nivelele de tensiune și puterea acestor semnale; secvența de evoluție în timp a acestor semnale în raport cu „ciclul mașină" etc.

Scanned with OKEN Scanner

Datorită acestei diversităţi de periferice şi semnale este necesar să se introducă între periferice şi unitatea centrală (procesor) un sistem intermediar care să realizeze următoarele funcţii: adaptarea nivelelor semnalelor (o adaptare tehnologică); adaptarea la formatul liniilor de legătură cu procesorul şi controlul funcţionării perifericelor (adică elaborarea sau interpretarea unor secvenţe cu evoluţie în timp ca semnale de control şi comandă a perifericelor).

Pe de altă parte, unitatea centrală de prelucrare are diferite moduri de schimb de informaţie cu perifericele (la nivelul 1 al inferfeţei — v. fig. 11. 26 şi 11. 32), cum sînt: legăturile programate, întreruperile prin apeluri perioritare şi accesul direct la memorie. Este evident că pentru fiecare mod de schimb de informaţie trebuie să existe un sistem intermediar specific de adaptare. Pentru a se evita de a avea pentru fiecare periferic atîtea sisteme de adaptare cîte moduri de legătură există, s-a introdus un al doilea nivel de interfaţă (nivelul 2 în figurile 11.26 şi 11.32). În acest fel, sistemul intermediar (interfaţa globală) a fost fracţionat în două părţi (v. fig. 11.32): o parte specifică fiecărui periferic de proces, dar aceeaşi pentru orice mod de legătură — aceasta fiind *interfaţa industrială* şi o parte specifică fiecărui mod de legătură cu unitatea centrală de prelucrare, indiferent de periferic — aceasta fiind *adaptarea unităţii centrale la interfaţa industrială* (v. fig. 11.32) ale cărei principii au fost prezentate în cadrul paragrafului precedent.

Astfel, adaptarea unităţii centrale la interfaţa industrială asigură următoarele funcţiuni:

— adresarea (apelul) perifericelor;
— condiţionarea semnalelor;
— transmiterea bidirecţională a informaţiei.

În sfîrşit, **interfaţa industrială** — care asigură adaptarea la fiecare periferic al procesului — este specifică fiecărui tip de periferic în parte. Adaptarea la perifericul de proces, asigurată de interfaţa industrială, constă din:

— memorarea informaţiei emise/transmise de/către perifericele de proces;
— controlul şi comanda secvenţială a schimburilor cu perifericele;
— adaptarea semnalelor: format, natură, nivel.

Ansamblul dispozitivelor fizice care realizează aceste funcţii ale interfeţei industriale pentru fiecare tip de periferic de proces se numeşte **cuplor.** O interfaţă industrială poate avea unul sau mai multe cuploare, fiecare dintre acestea avînd o anumită autonomie de funcţionare în comparaţie cu „adaptarea unităţii centrale la interfaţa industrială" (adică: unitatea centrală, prin intermediul „adaptarea unităţii centrale la interfaţa industrială", cere ansamblului cuplor-periferic un schimb unitar de informaţie, ca de exemplu citirea unui caracter; cuplorul — după ce a primit acest „ordin" — îl va executa prin perifericul indicat, în mod cu totul independent faţă de unitatea centrală).

Cuploarele, realizate în concordanţă cu specificul diferitelor tipuri de periferice de proces, sînt şi ele de mai multe feluri: cuploare pentru senzorii analogici (numite şi module sau dispozitive pentru intrările analogice); cuploare pentru senzorii numerici (numite şi module sau dispozitive pentru intrările numerice); cuploare pentru executori numerici, analogici sau combinate (numite şi module sau dispozitive pentru ieşirile numerice sau/şi analogice).

Scanned with OKEN Scanner

Structura interfeţei industriale. Interfaţa industrială este un echipament cu o structură modulară, fiind alcătuită din cîteva dispozitive funcţionale distincte (amplificatoare, convertoare, multiplexoare, decodoare, blocuri de comandă etc.). Considerînd ca sens de referinţă pentru transferul informaţiei, sensul proces → unitate centrală, atunci modulele structurale, din care este alcătuită o interfaţă industrială, sînt următoarele: cuplorul pentru intrările analogice; cuplorul pentru intrările numerice; cuplorul pentru ieşirile analogice şi cuplorul pentru ieşirile numerice. În majoritatea cazurilor, dispozitivele interfeţei prin care se transmit semnalele de comandă către proces („ieşiri") sînt asociate într-un singur cuplor: dispozitivul pentru ieşirile numerice şi analogice (v. *Ex. 11—1* şi fig. 11.3).

Cuplorul pentru intrările analogice îndeplineşte următoarele funcţiuni: „administrarea" funcţionării unui anumit număr de senzori analogici din proces (de exemplu: 128 senzori/modul cu 64 module/cuplor); primirea semnalelor analogice prelevate din proces; transformarea şi normalizarea semnalelor la nivelul impus de dispozitivele de amplificare şi conversie; conversia semnalelor analogice în semnale numerice şi transferul informaţiei către unitatea de prelucrare (la nivelul 2 al interfeţei).

Configuraţia generală a unui modul pentru intrările analogice este indicată în schema de figura 11.33.

Dispozitivul de multiplexare primeşte semnalele analogice transmise de senzorii din proces şi le transmite dispozitivului de conversie analog-

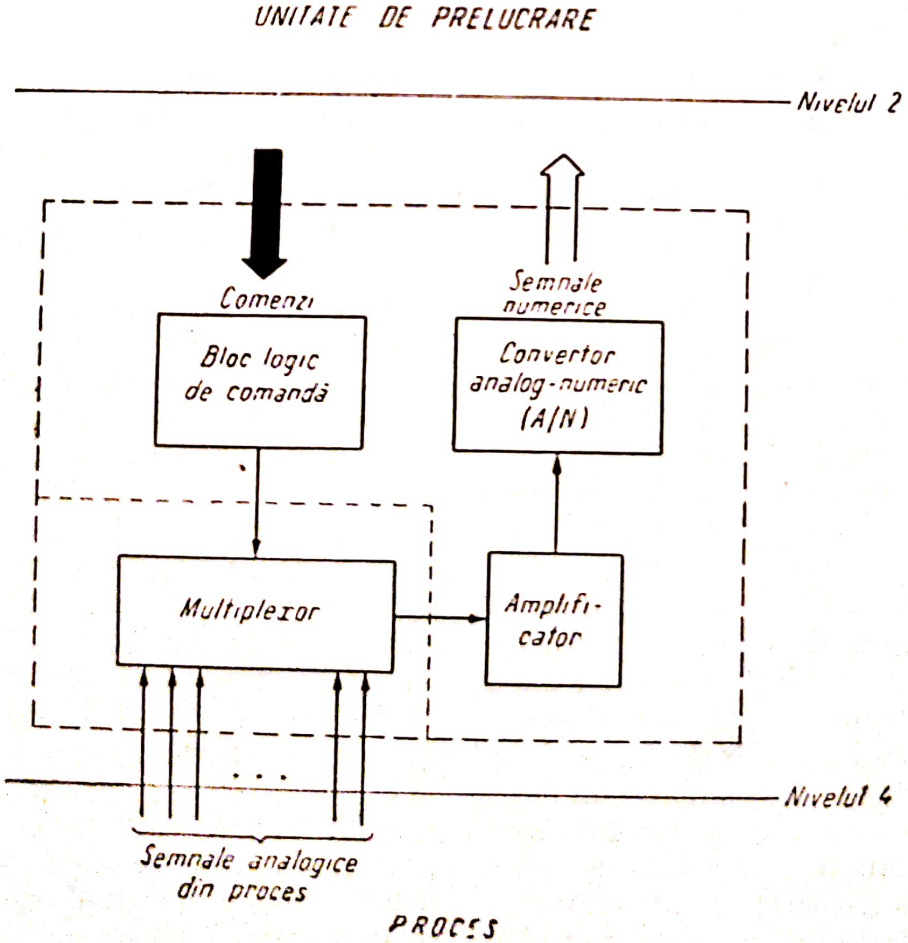


Fig. 11.33.

Scanned with OKEN Scanner

numeric. Multiplexorul realizează şi o anumită succesiune de „tratare" (administrare) a senzorilor, printr-o matrice de comutare comandată prin „blocul logic de comandă" (v. fig. 11.33).

Dispozitivul „convertor analog-numeric" (v. fig. 11.33), precedat de un amplificator cu coeficient de amplificare comandat, trebuie să asigure o rată de conversie în concordanţă cu viteza de variaţie a semnalului analogic prelevat din proces.

Semnalele numerice obţinute după conversie sînt „depuse" în nişte registre tampon plasate dincolo de nivelul 2 al interfeţei. Conţinutul acestor registre este transferat, la comandă, unităţii centrale de prelucrare, prin una din cele trei modalităţi (legătură programată, intrare prioritară prin întreruperi sau acces direct memorie).

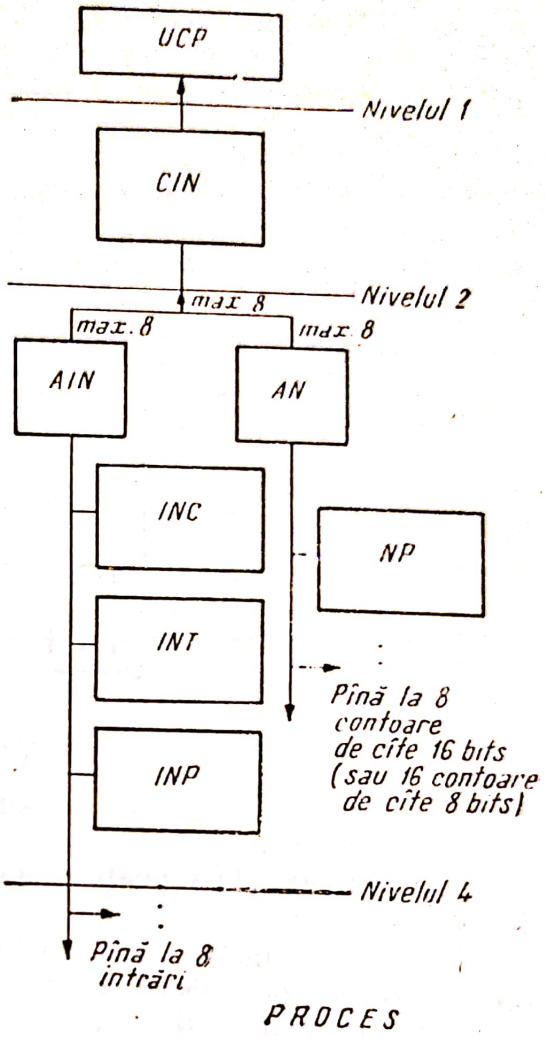


Fig. 11.34.

Cuplorul pentru intrările numerice. El asigură recepţionarea în timp real a semnalelor numerice. În componenţa acestor cuploare intră numeroase module, aşa cum se arată schematic în figura 11. 34, dintre care esenţiale sînt următoarele : intrările statice pentru senzorii cu contact (*INC*), intrările statice şi dinamice pentru semnalele în formă de treaptă de tensiune (*INT*), intrările dinamice pentru semnalele formate din pulsuri în grupuri repetate (*INP*), adaptoarele intrărilor numerice (*AIN*), numărătoarele de pulsuri (*NP*), adaptoarele pentru numărătoarele de pulsuri (*AN*) şi controlul intrărilor numerice (*CIN*). În schema din figura 11.34 este indicată şi unitatea centrală de prelucrare (*UCP*).

De obicei, informaţia numerică este organizată pe grupe de cîte 16 biţi, formatul fiind în funcţie de tipul aplicaţiei : biţi independenţi (nivele de tensiune, contacte), numere binare, caractere codificate în binar.

Cuploarele sînt modularizate (de exemplu, pot avea 8 adaptoare fiecare cu cîte 8 intrări, iar o interfaţă poate avea pînă la 8 astfel de cuploare). În cazul acestui exemplu, transferul de informaţie la nivelul 2 al interfeţei se face prin 8.8.16 = 1 024 biţi/cuplor.

„Controlul intrărilor numerice" (*CIN* în fig. 11.34) acceptă comenzile de control şi citire ale unităţii centrale de prelucrare şi acţionează conform descrierii de detaliu prezentată la *Ex. 11—10*.

Adaptorul intrărilor numerice (*AIN* în fig. 11.34) este utilizat pentru organizarea transferului de date primite prin semnalele de la intrările

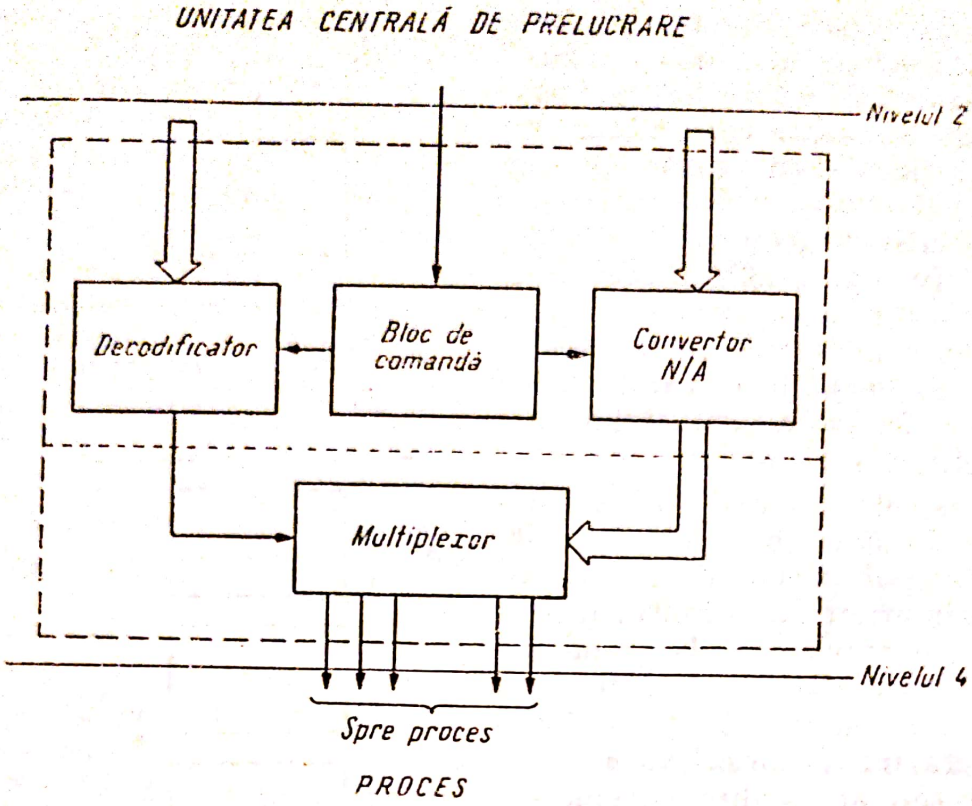


Fig. 11.35.

numerice. Fiecare *AIN* poate prelua un anumit număr de bits (de exemplu 128 bits).

Adaptorul numărătoarelor (*AN* în fig. 11.34) este destinat organizării numărătoarelor de pulsuri de la intrările numerice (*NP*). Fiecare *AN* are o anumită capacitate de numărare (de exemplu 2^7 bits), cu care asigură preluarea conținutului numărătoarelor de pulsuri de la intrările numerice *NP* (care au capacități de 2^3 sau 2^4 bits).

Cuploarele pentru ieșirile analogice. Ele au rolul de a transmite spre proces (la executorii analogici) un semnal de comandă sub formă analogică.

Structura generală a unui asemenea cuplor este reprezentată în figura 11.35.

Cuploarele pentru ieșirile numerice. Aceste cuploare asigură transmiterea spre proces (la executorii numerici) a unor comenzi în formă de semnale discrete ca: pulsuri individuale de tensiune (bits „izolați"), trenuri de pulsuri, semnale binare pe linii paralele (de exemplu semnale cu 8 poziții binare — prin 8 fire paralele). Ele sînt formate din relee cu contacte (de obicei cu mercur), circuite de comutare cu semiconductori, registre cu celule binare și matrice cu contacte.

Exemple de interfețe industriale

Datorită varietății de periferice de proces, de structuri ale semnalelor și condiții specifice proceselor tehnologice, numărul și tipul modulelor de intrare/ieșire (cuploare) ale interfețelor industriale este foarte divers, ca și tipul interfețelor însăși.

Scanned with OKEN Scanner

În aceste condiţii, pentru a familiariza cititorul cu aparatură utilizată în ţara noastră vom prezenta, în continuare, două exemple concrete : cuplorul sistemelor de fabricaţie românească FELIX C 32P şi interfaţa standard IBM 1827.

Ex. 11—9. **Cuplorul de proces al sistemului FELIX C—32P.** Structura acestui cuplor a fost prezentată în figura 11.3 (v. *Ex. 11—1*). În cadrul acestui exemplu vom analiza succint fiecare dispozitiv de bază al acestui cuplor (v. *DIN*, *DIA* şi *DENA* din figura 11.3).

Dispozitivul pentru intrările numerice (DIN). Acest dispozitiv primeşte semnalele numerice prelevate din proces şi le transformă în semnale compatibile cu unitatea centrală FELIX C—32, căreia îi sînt transmise prin legăturile programate, sub controlul instrucţiunii RD (v. *Ex. 11—4*). În acelaşi timp, *DIN* centralizează cererile de întrerupere venite din proces, ordonîndu-le după priorităţi şi transmiţîndu-le unităţii centrale. Pe lîngă acestea, *DIN* asigură şi legătura între unitatea centrală de prelucrare şi consola operatorului de proces (*COP* — v. fig. 11.3).

DIN poate prelua semnale numerice de forma : nivele de tensiune, pulsuri izolate, trenuri de pulsuri repetate, pulsuri de numărare şi semnale de întrerupere. Trebuind să primească astfel de semnale, în structura dispozitivului de intrări numerice, reprezentată prin schema funcţională din figura 11.36, intră următoarele părţi componente : un ansamblu de circuite de comandă (format din : *CI* — circuitele interfaţă/calculator, *DAD* — blocul decodor al adresei dispozitivelor, *AM—2* — blocul de acceptare martor, *DAL* — blocul decodor al adresei liniilor, *DNR* — blocul pentru dispozitivele nerecunoscute şi blocul *SAU* cu 64 × 16 intrări) ; circuite de adaptare pentru intrările numerice formate din : *AINC* — adaptoare pentru intrările prin contacte, *AINT* — adaptoare pentru intrările prin nivel de tensiune, *AINR* — adaptoare pentru intrările prin trenuri de pulsuri repetate ; *AINP* — adaptoarele intrărilor de numărare a pulsurilor) ; blocuri de numărare a pulsurilor (*NP*) şi blocuri pentru semnalele de întrerupere din proces (*BI*).

Circuitele interfaţă/calculator (*CI*) sînt circuite de adaptare : amplificatoare emiţătoare de semnale către interfaţă şi amplificatoare receptoare pentru semnalele rezultate din interfaţă. *CI* transmite către *DIN* următoarele semnale : $\overline{AOR}$ la $\overline{A15R}$ care reprezintă codul adresei; $\overline{SP}$ — martorul unităţii centrale ; $\overline{H1M}$ — semnalul de tact (cu frecvenţa de 1 MHz) ; $\overline{RAZ}$ — semnalul de iniţializare şi $\overline{RWD}$ — semnalul pentru indicarea sensului de transfer al informaţiei.

Circuitele interfaţă/calculator (*CI*) primesc de la celelalte blocuri ale *DIN*, pentru a le transmite apoi unităţii centrale (*UC* — v. fig. 11.36), următoarele semnale : *CC3* — un indicator de stare a dispozitivelor cu semnificaţia „dispozitiv nerecunoscut" ; *CC4* — indicatorul stării „dispozitiv ocupat" ; *ASP* — care indică faptul că instrucţiunea transmisă de unitatea centrală a fost luată în considerare ; *DOR* la *D15R* — cele 16 semnale numerice purtătoare de date cu privire la proces ; *NCI* — semnale reprezentînd nivelele cererilor de întreruperi.

Semnalele *L0*, *L1*, ..., *L63* sînt produse la ieşirea blocului decodor pentru adresa liniilor (*DAL* în fig. 11.36), prin decodificarea semnalelor

Scanned with OKEN Scanner

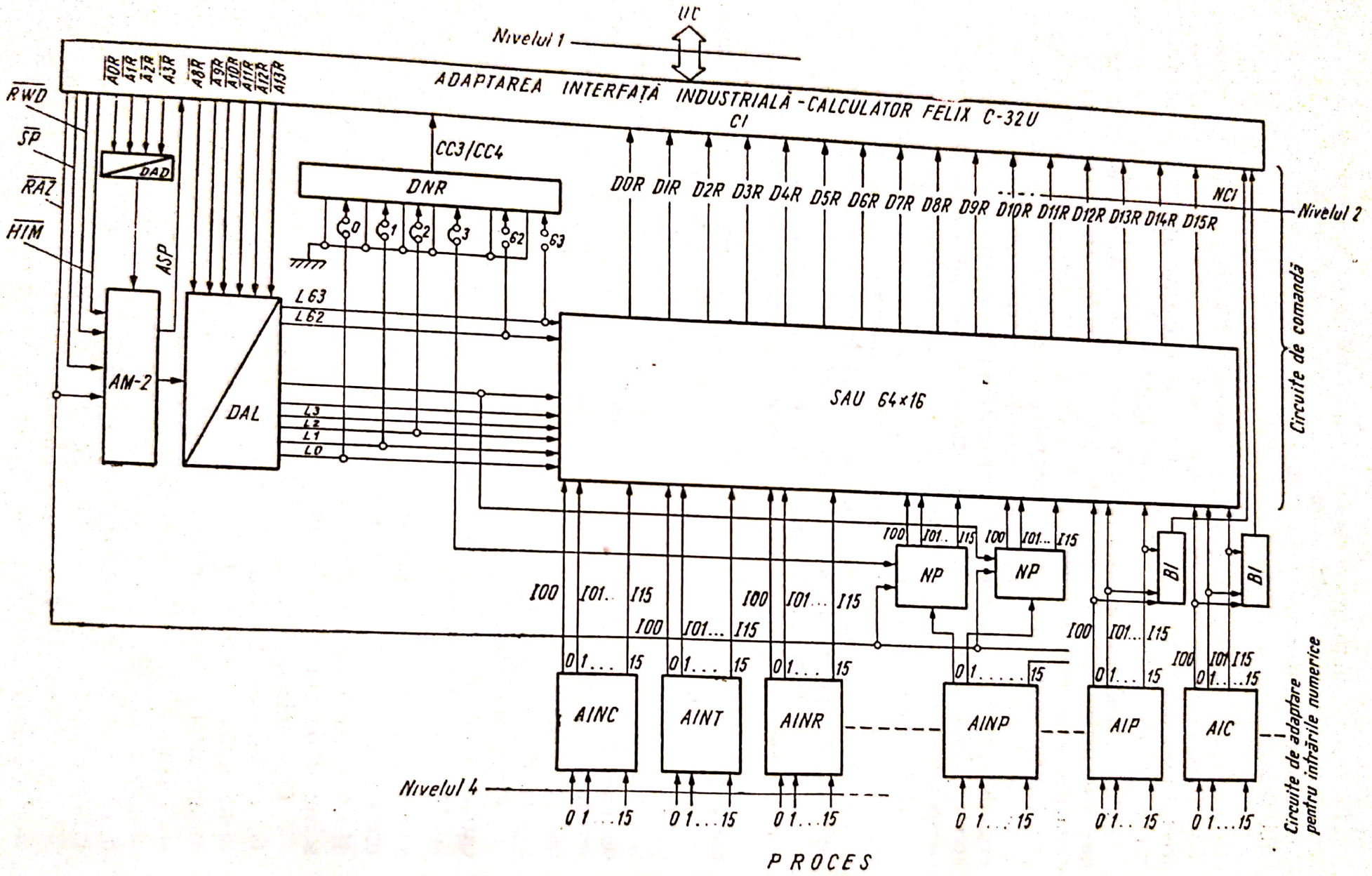

UC
Nivelul 1
ADAPTAREA INTERFAȚĂ INDUSTRIALĂ - CALCULATOR FELIX C-32U
CI
RWD
SP
RAZ
HIM
DAD
ASP
AM-2
DAL
CC3/CC4
DNR
L63
L62
L3
L2
L1
L0
SAU 64×16
NCI
Nivelul 2
Circuite de comandă
NP
NP
BI
BI
AINC
AINT
AINR
AINP
AIP
AIC
Nivelul 4
Circuite de adaptare pentru intrările numerice
PROCES

Fig. 11.36.

$\overline{A8R}$, ..., $\overline{A13R}$ transmise de unitatea centrală prin *CI*, şi indică selectarea uneia din cele 64 linii de intrare în blocul *SAU*.

La fiecare din aceste 64 linii se poate conecta unul din următoarele blocuri: adaptor pentru intrările numerice (*AINC* sau *AINT* sau *AINR*), blocul de numărare a pulsurilor (*NP*) şi blocul de întreruperi de la proces sau de la alte dispozitive.

Pot exista cel mult 27 linii afectate întreruperilor şi anume: linii pentru întreruperi de la proces (prin adaptoarele de întreruperi proces *AIP*); linii de întreruperi de la consola operatorului de proces (prin adaptorul *AIC*) şi linii pentru întreruperi de la dispozitivele şi semnalele de stare ale *DIA* şi *DENA* (v. fig. 11.3).

După cum se vede din schema *DIN* (fig. 11.36), blocul *SAU* primeşte semnalele *I00*, ..., *I15* de la toate intrările numerice ale cuplorului *DIN* şi le transmite unităţii centrale de prelucrare, prin intermediul *CI*, sub forma semnalelor *D0R*, *D1R*, ..., *D15R* corespunzător liniei selectate (de semnalele $Li =$ „1" $i = 1,2, \ldots, 64$) prin adresa indicată de instrucţiune. Deoarece adaptarea interfaţă industrială — calculator FELIX C—32U (notată cu *CI* în figura 11.36) este comună pentru toate dispozitivele conectate la legătura programată, fiecare semnal de la *CI* (la nivelul 1 al interfeţei) reprezintă suma logică a semnalelor corespunzătoare emise de fiecare dispozitiv din proces în parte; deci emisia trebuie să se facă numai de către dispozitivul adresat de instrucţiune, însoţită şi de către un semnal de validare.

Adaptorul pentru intrările prin contacte (*AINC* în fig. 11.36) poate primi 16 semnale de la proces sub forma de „contact închis" sau „contact deschis" pe care le transformă în semnale logice standard (TTL) conform convenţiei: contact închis = semnal logic „1" şi contact deschis = semnal logic „0". Aceste semnale (*I00*, *I01*, ..., *I15* în figura 11.36) sînt transmise unităţii centrale prin blocul *SAU* şi adaptarea *CI* sub controlul ansamblului circuitelor de comandă. *AINC* este dotat şi cu filtre de zgomot.

Adaptorul pentru intrările prin nivel de tensiune (*AINT* — fig. 11.36) poate primi 16 semnale de la proces sub forma unor trepte de tensiune electrică, pe care le transformă în semnale logice (TTL) conform convenţiei: — 30V la — 6V = semnal logic „0" şi — 1 V la + 30 V = semnal logic „1". Aceste semnale (*I00*, *I01*, ..., *I15* în figura 11.36) sînt transmise unităţii centrale sub controlul ansamblului de comandă al *DIN*. Şi acest adaptor *AINT* asigură filtrarea semnalelor perturbatoare (a zgomotelor care apar pe liniile de intrare).

Adaptoarele pentru intrările prin trenuri de pulsuri repetate (*AINR* — fig. 11.36) se deosebesc de *AINT* — numai prin eliminarea condensatorului de filtrare trece sus, asigurînd astfel o viteză de transmitere mai mare (cu frecvenţa de 100 kHz).

Adaptoarele de intrare pentru numărarea pulsurilor (*AINP* — fig. 11.36) primesc pulsurile venite de la proces şi le transformă în pulsuri compatibile *TTL* folosite pentru comanda blocurilor de numărare (*NP* — fig. 11.36). *AINP* are aceeaşi schemă ca şi *AINT*, cu deosebirea că valoarea capacităţii condensatorului de filtrare este mai mică, pentru a nu depăşi frecvenţa maximă admisibilă a pulsurilor numărate care este de 5 kHz. Blocul de numărare (*NP*) este, de fapt, un numărător cu 16 poziţii binare, deci cu capacitatea de $2^{16} - 1 = 65\,535$. Numărătorul este şters prin

Scanned with OKEN Scanner

semnalul *RAZ* sau după terminarea unei instrucțiuni prin care s-a citit starea numărătorului.

Dispozitivul pentru intrările analogice (*DIA* — v. fig. 11.3). Acest cuplor asigură legătura între senzorii analogici din proces (de la nivelul 4 al interfeței) la „adaptarea interfață industrială — calculator FELIX C—32U" (notată cu *CI* în figura 11.37), adică la nivelul 2 al interfeței.

Comunicarea între nivelele 1 și 2 ale interfeței se face prin adaptarea *CI* (v. fig. 11.37). Această asigură unității centrale (*UC* — fig. 11.37) realizarea recepționării/transmiterii informației direct de la cuplorul *DIA* prin legătura programată. Utilizînd cele două instrucțiuni de scriere directă WD și de citire directă RD (v. *Ex. 11—4*) se asigură selectarea punctelor de măsurare (prin instrucțiunea WD) și introducerea în memoria calculatorului a informației prelevate (prin instrucțiunea RD). Pentru aceasta, adaptarea CI cuprinde:

16 fire de adresă *A0 ÷ A15* (v. fig. 11.37) repartizate funcțional astfel: *A0 ÷ A3* — adresa dispozitivului, *A4 ÷ A7* — adresa multiplexorului, *A8 ÷ ÷ A13* — adresa liniei și *A14 ÷ A15* neocupate;

32 fire bidirecționale de informație (*D0 ÷ D5*) — semnul *SING* al mărimii analogice convertite numeric, *D6 ÷ D15* — expresia binară a mărimii analogice convertită numeric, *D16 ÷ D31* — neutilizate);

un fir pentru semnalul martor al unității centrale (*SP*);

un fir pentru semnalul de acceptare martor (*ASP*);

două fire de înscriere a indicatorilor de stare ai dispozitivelor (*CC3* — — dispozitiv nerecunoscut și *CC4* — dispozitiv ocupat);

un fir pentru semnalul *RWD* care indică sensul schimbului de informație;

un fir pentru semnalul de punere la zero *RAZ*, care asigură și inițializarea dispozitivului și

un fir pentru semnalul de tact *H1M*.

Modul de lucru este următorul:

UC transmite o instrucțiune WD pentru selectarea punctului de măsurare din proces. Partea de adresă a instrucțiunii WD realizează selectarea dispozitivului pe firele *A0 ÷ A3*, selectarea multiplexorului pe firele *A4 ÷ A7* și selectarea liniei din multiplexor pe firele *A8 ÷ A13* (v. fig. 11.37). Partea de adresă a instrucțiunii WD devine acceisbilă *DIA* numai după ce *UC* a emis semnalul martor *SP*;

după recunoașterea adresei dispozitivului, blocul de comandă *DIA* validează înscrierea și decodificarea adreselor de multiplexor și de linie (adică adresa punctului de măsurare din proces), după care formează și transmite semnalul martor *ASP* sau unul din indicatorii de stare ai dispozitivului (*CC3/CC4*);

la sfîrșitul instrucțiunii WD, dispozitivul trece în starea OCUPAT și legătura cu *UC* se întrerupe. La nivelul *DIA*, însă, decodificarea adreselor de linie și de multiplexor determină acționarea elementului de multiplexor rezultat și declanșarea, cu o întîrziere de 3 ms, a convertorului analog-numeric;

la sfîrșitul conversiei analog-numerice, convertorul suprimă comanda de acționare a releului multiplexor și, după 5 ms, declanșează un semnal de întrerupere (*ICAN*);

Scanned with OKEN Scanner

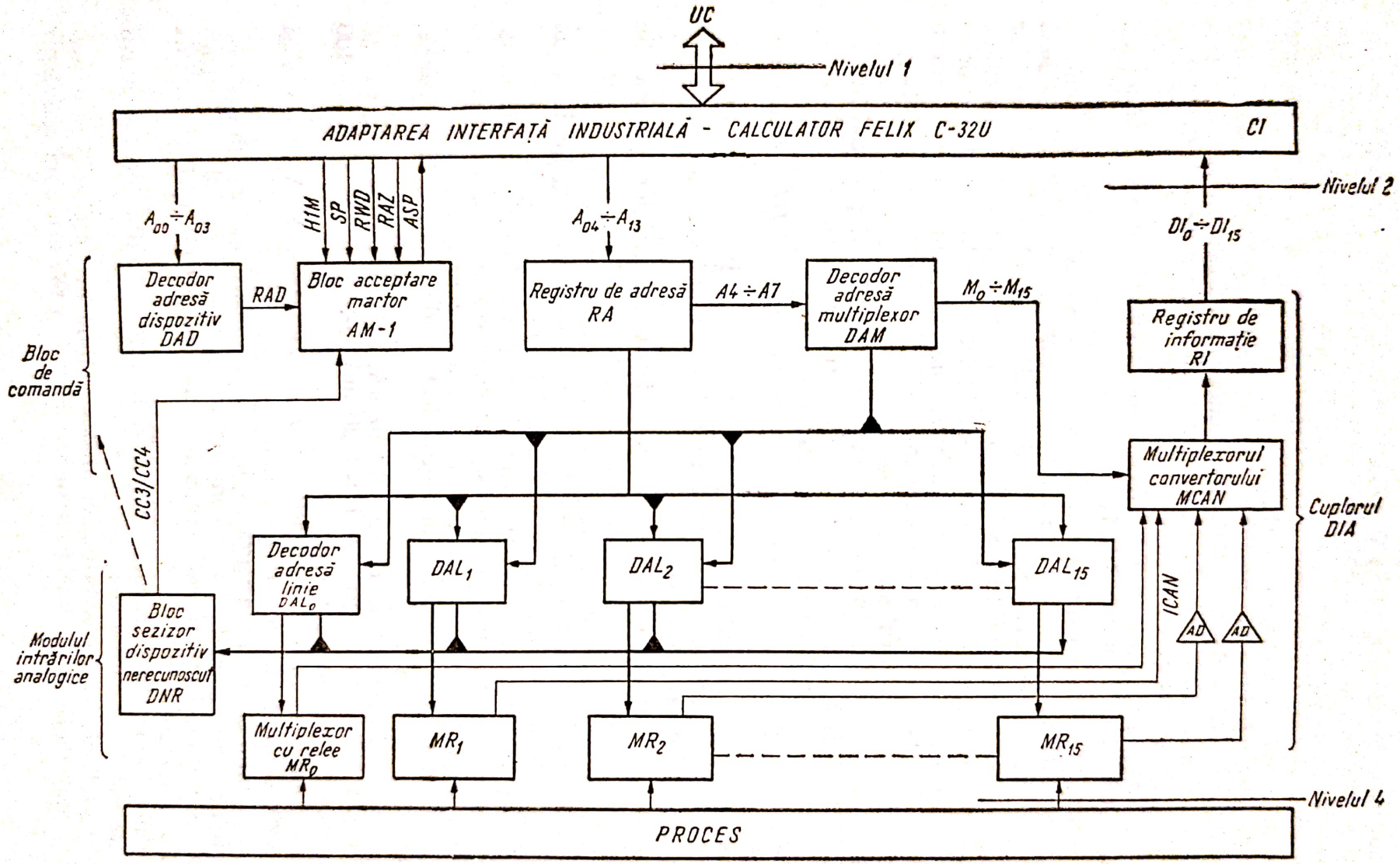

UC
Nivelul 1
ADAPTAREA INTERFAȚĂ INDUSTRIALĂ - CALCULATOR FELIX C-32U
CI
Nivelul 2
$A_{00} \div A_{03}$
HIM
SP
RWD
RAZ
ASP
$A_{04} \div A_{13}$
$DI_0 \div DI_{15}$
Decodor adresă dispozitiv DAD
RAD
Bloc acceptare martor AM-1
Registru de adresă RA
A4 ÷ A7
Decodor adresă multiplexor DAM
$M_0 \div M_{15}$
Registru de informație RI
Bloc de comandă
CC3/CC4
Multiplexorul convertorului MCAN
Cuplorul DIA
Decodor adresă linie DAL_0
DAL_1
DAL_2
DAL_{15}
ICAN
AD
AD
Modulul intrărilor analogice
Bloc sezizor dispozitiv nerecunoscut DNR
Multiplexor cu relee MR_0
MR_1
MR_2
MR_{15}
Nivelul 4
PROCES

Fig. 11.37.

această întrerupere *ICAN* solicită sistemului de programe lansarea instrucțiunii RD în vederea citirii valorii măsurate (a introducerii în memoria calculatorului). La instrucțiunea RD se folosesc numai firele *D0 ÷ D15* prin care se transmite valoarea convertită numeric a mărimii analogice selectate prin instrucțiunea WD;

transferul informației numerice are loc după recepționarea semnalului martor *SP* al *UC* și după ce *DIA* și-a recunoscut adresa (transferul se face între registrul de informație al *DIA* și registrul din *UC* definit de instrucțiunea RD prin firele *A0 ÷ A3*);

În final, instrucțiunea RD trece dispozitivul cu care s-a lucrat din starea OCUPAT în starea LIBER.

Revenind asupra cuplorului *DIA*, structura sa este redată schematic în figura 11.37 din care rezultă că *DIA* este format dintr-un *bloc de comandă* (alcătuit din: „adaptarea interfață industrială — calculator" *CI*; decodor adresă dispozitiv — *DAD*; blocul acceptare martor — *AM—1*; registrul de adresă — *RA*; registrul de informație — *RI*; decodor adresă multiplexor — *DAM*; blocul de sesizare a dispozitivului nerecunoscut — *DNR*; multiplexorul convertorului analog-numeric — *MCAN* care — pe figură — înglobează și convertorul analog-numeric, ce va fi descris separat — v. fig. 11.38) și *modulul intrărilor analogice* (alcătuit din: decodor adresă linie — — *DAL0, DAL1, ..., DAL15*; multiplexorul cu relee — *MR0, MR1, ..., MR15* și amplificatoare diferențiale — *AD*).

Funcțiile principale pe care le realizează cuplorul *DIA* sînt următoarele:

conexiunea prin adaptarea *CI* la calculatorul FELIX C—32U (cu; legătura programată, recunoașterea adresei senzorului selectat, alcătuirea unor mesaje de transfer etc.);

adaptarea semnalelor analogice în vederea aducerii lor în una din gamele de intrare ale convertorului analog-numeric;

reducerea semnalelor perturbatoare și filtrarea semnalului util;

amplificarea semnalelor de nivel mic în vederea aducerii lor în gama de lucru a convertorului (0 la 4 V sau 1 la 5 V);

multiplexarea semnalelor analogice în scopul reducerii numărului de amplificatoare și a utilizării unui singur convertor analog-numeric;

conversia semnalelor analogice în semnale numerice.

DIA poate primi semnale analogice electrice de curent și de tensiune, în următoarele game:

— pentru semnalele de curent: $2 \div 10$ mA, $1 \div 5$ mA, $4 \div 20$ mA, $10 \div 50$ mA, $0 \div \pm 5$ mA, $0 \div \pm 10$ mA, $0 \div \pm 20$ mA, $0 \div \pm 50$ mA și $0 \div \pm 100$ mA, toate în curent continuu;

— pentru semnalele de tensiune: $0 \div \pm 1$V, $0 \div \pm 10$ V, $1 \div 5$ V și $0 \div \pm 20$ V, toate în curent continuu.

Un modul de intrare asigură cuplarea a 64 senzori analogici din proces, iar un bloc de comandă poate controla maximum 16 module de intrări analogice (prin urmare *DIA* are o capacitate minimă de 64 intrări și maximă de 1 024 intrări).

Dintre blocurile structurale ale *DIA* le vom prezenta succint numai pe cele mai reprezentative:

— multiplexorul convertorului analog — numeric (*MCAN*) care realizează multiplexarea modulelor de intrare în scopul folosirii unui convertor

Scanned with OKEN Scanner

analog-numeric unic. Acest multiplexor conține cîte un releu pentru fiecare modul de intrare, releu care basculează ori de cîte ori este selectat cîte un punct conectat la modulul de intrare. Semnalele *M0, ..., M3* care selectează releele multiplexorului convertorului sînt furnizate de blocul *DAM* (v. fig. 11.37);

— multiplexorul cu relee (*MR0, MR1, ..., MR15*) permite conectarea pe rînd la convertorul analog-numeric a cîte unui canal de intrare din cele 64 canale posibile. Multiplexorul este realizat cu 64 relee cu contacte cu mercur, model miniatură, fiecare avînd 2 contacte-comutatoare;

— *convertorul analog-numeric*, a cărui schemă de principiu este prezentată în figura 11.38, are următoarele caracteristici tehnice: gamele de măsură pentru semnalele de intrare $0 \div \pm 4$ V și $1 \div \pm 5$ V, timpul de conversie mai mic decît 190 μs, eroarea de conversie $\pm 0{,}1\%$ din valoarea efectiv măsurată plus 0,05% din valoarea maximă (4V) la $23°C \pm 2°C$, semnalul numeric de ieșire cu 10 bits plus semn (TTL), emiterea unui semal de defect dacă la operațiile de control se depășește o eroare absolută de $\pm$ 10 mV.

În cadrul *DIA*, acest convertor este „amplasat" între multiplexorul convertorului *(MCAN)* și registrul de informație — *RI* (v. fig. 11.37).

Principiul de funcționare al convertorului analog-numeric se bazează pe metoda aproximațiilor succesive. Circuitele lui componente se pot grupa astfle (v. fig. 11.38): un bloc analogic (format dintr-un amplificator de intrare, un comparator și trighere); un bloc logic, de comandă a conversiei, dotat și cu un registru pentru păstrarea informației convertite; un bloc de conversie numerică — analogică; un bloc logic de control al preciziei și de transmisie a informației convertite; un bloc cu surse electrice de c.c. (pentru producerea tensiunii de control — U_C, tensiunii de referință — U_R și a unor tensiuni de alimentare).

La intrarea comparatorului se aplică, pe de o parte, tensiunea U_x rezultată din adaptarea semnalului analogic de intrare prelevat din proces și, pe de altă parte, tensiunea de reacție U_R. Această tensiune U_R este formată din trepte ponderate poziționat în cod binar, în mod descrescător, care sînt acceptate sau respinse, după cum suma lor este mai mică sau — respectiv — mai mare decît tensiunea de intrare ce trebuie convertită U_x. Treptelor acceptate le corespund stări „1", iar celor respinse „0", valori logice transmise printr-un tren de pulsuri *C* (convertite) în registrul convertorului în care, la sfîrșitul conversiei, se găsește valoarea numerică binară, codificată, a mărimii analogice.

În ceea ce privește notațiile semnalelor din figura 11.38, acestea au următoarele semnificații:

H1M este semnalul de tact (cu frecvența de 1 MHz) emis de adaptarea *CI* (v. fig. 11.37);

$\overline{B1}$, $\overline{B2}$, ..., $\overline{B10}$ sînt pulsuri (semnale binare cu valori logice „0" sau „1") care, codificat, reprezintă valoarea transmisă de semnalul analogic. Ele provin din registrul convertorului și sînt introduse simultan la intrările convertorului numeric-analog și blocului de control al preciziei conversiei, de la ieșirea căruia sînt trimise — în caz de validare a conversiei — în registrul de informație *RI* al *DIA* (v. fig. 11.37);

$\overline{SGN}$ este un semnal binar pentru indicarea semnalului valorii numerice obținută prin conversie;

Scanned with OKEN Scanner

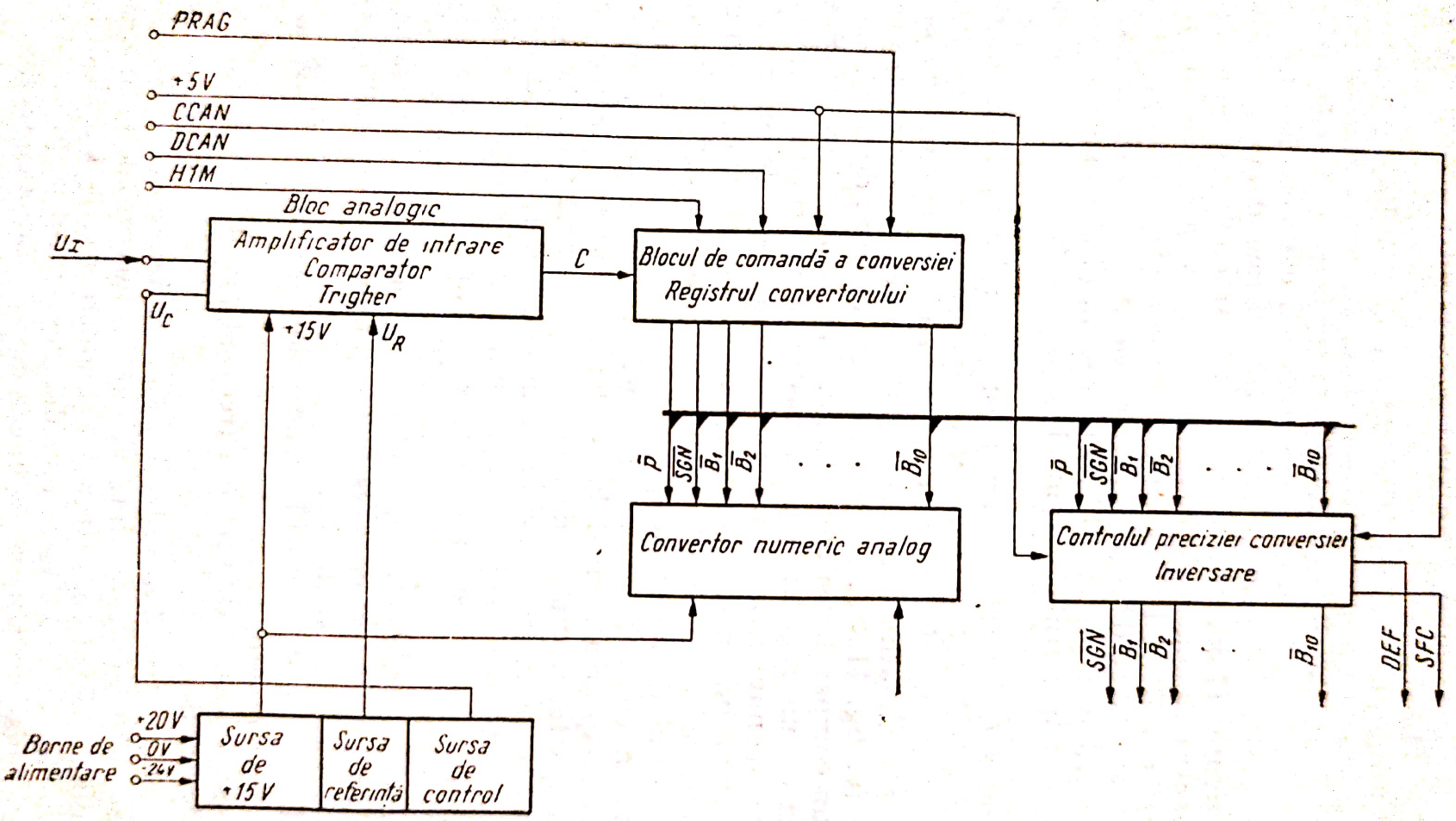


Fig. 11.38.

Scanned with OKEN Scanner

$\overline{P}$ — semnal binar de comandă a controlului conversiei;

DCAN — semnal de declanşare a convertorului analog-numeric (este emis de blocul acceptare martor *AM*—1(v. fig. 11.37);

SFC — semnalul „sfîrşitul conversiei", produs după terminarea conversiei şi controlului preciziei ei (care, în total, durează 190 μs);

PRAG — semnal care comandă convertorului analog — numeric alegerea gamei de funcţionare (1—5 V pentru *PRAG* = „1" şi 0—4 V pentru *PRAG* = „0"). Acest semnal este emis de către decodorul adresă multiplexor — *DAM*(v. fig. 11.37);

CCAN — semnal pentru comanda/controlul conversiei. Este emis de către decodorul adresă linie (*DAL* — v. fig. 11.37). În cazul blocului *DAL* 2 semnalul de selecţie al liniei *LO* va fi transmis şi convertorului analog — numeric (sub numele de *CCAN*). În acest fel, ori de cîte ori se va selecta linia zero din multiplexorul *MR2* (v. fig. 11.37), convertorul analog-numeric va primi şi comanda de control conversie şi asta deoarece la linia *LO* din multiplexorul 2 este conectată sursa de control a convertorului analog — numeric;

DEF — semnalul cu semnificaţia „convertor defect", el va fi emis numai dacă — în urma unei conversii de control — se constată o eroare mai mare decît ± 10 mV.

Dispozitivul de ieşiri numerice şi analogice (*DENA* în fig. 11.3). El asigură semnalele electrice, analogice şi numerice, pentru comanda unor executori din proces ca: contactoare de pornire/oprire a motoarelor electrice (mecanisme de execuţie), motoare pas cu pas, relee (cu sau fără autoreţinere), registre de mare viteză ale diverselor dispozitive, aparate de măsurat (cu afişare analogică sau numerică) etc.

Structura acestui cuplor *(DENA)* este redată prin schema de principiu din figura 11.39. După cum rezultă din această figură, ieşirile *DENA* sînt realizate prin următoarele module:

— modulul cu ieşiri pe contacte de relee *(MIREL)* care primeşte: 16 semnale informaţionale; un semnal de validare a informaţiei; un semnal de linie *iL*, $i = 00,01,\ldots,63$ (*iL* = „1" pentru modulul adresat căruia îi sînt destinate datele); semnalul de iniţializare — *RZMITI* şi semnalul de comandă. Datele *(DOO÷D15)* sînt memorate, în urma unei instrucţiuni *WD*, într-un registru unde sînt păstrate pînă la o nouă comandă de iniţializare (*RZMITI* = „1") sau pînă la recepţionarea semnalului de comandă. Sarcina admisă pe contactul releului este de 100 VA (2 A/50 V);

— modulul cu ieşiri pe amplificatoare *(MIA)* care primeşte: 16 semnale informaţionale; un semnal de validare a informaţiei; un semnal de linie *iL* ($i = 00,01,\ldots,63$) şi semnalul de iniţializare — *RZ*. Datele *(DOO÷D15)* sînt memorate într-un registru, în urma unei instrucţiuni *WD*, unde sînt păstrate pînă la o nouă comandă de înscriere destinată modulului sau pînă la o comandă de iniţializare. La ieşire din acest modul este furnizat un semnal numeric care poate comanda sarcini de maximum 3 W;

— modulul de ieşiri pe registru *(MIREG)* care primeşte de la ansamblul de comandă: 16 semnale informaţionale; un semnal de validare a informaţiei; un semnal de linie *iL* ($i = 00,01,\ldots,63$); semnalul de ini-

Scanned with OKEN Scanner

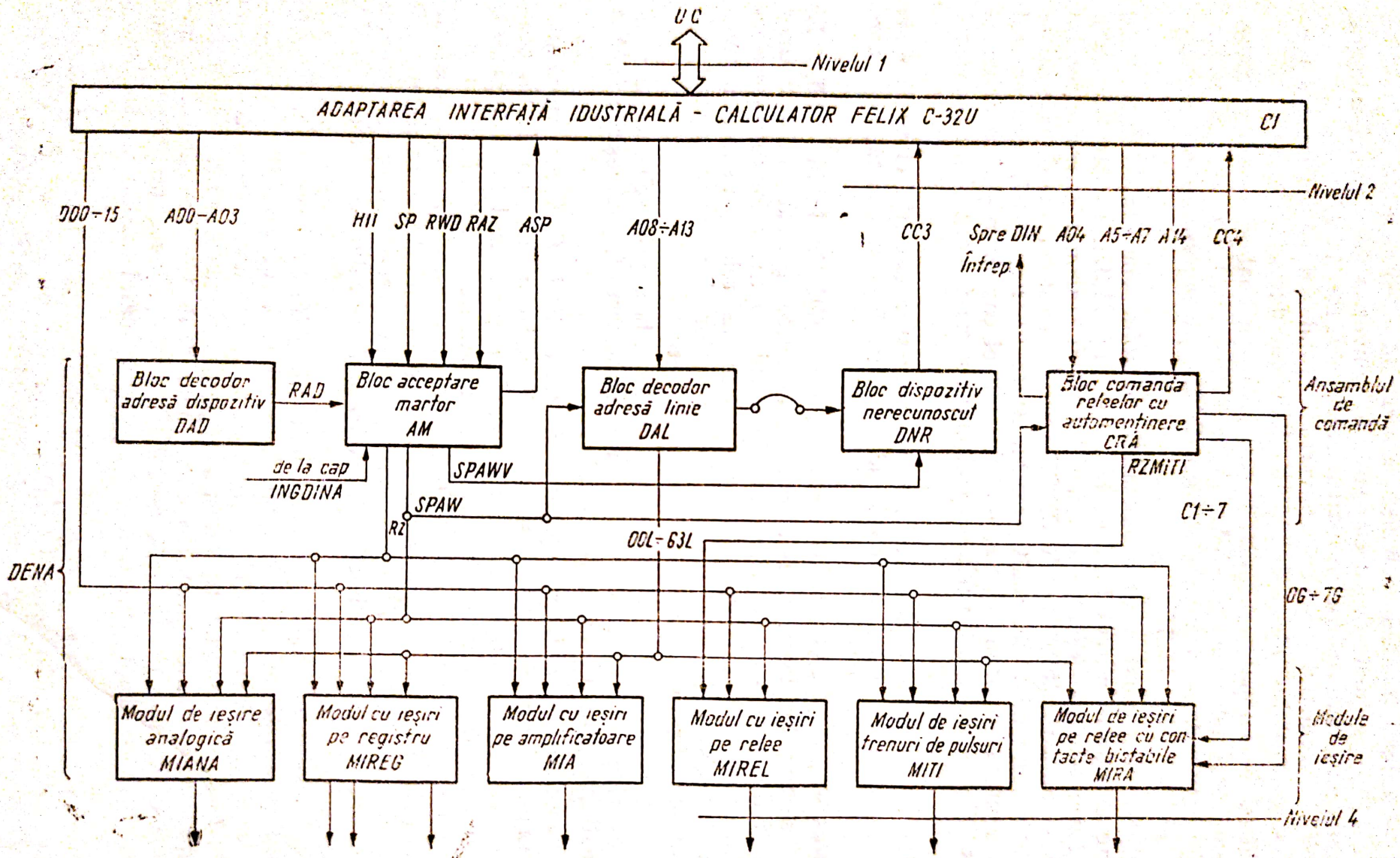


Fig. 11.39.

Scanned with OKEN Scanner

ţializare — *RZ* şi un semnal de validare a ieşirii. Datele *(D00÷D15)* sînt păstrate, între două comenzi, într-un registru tampon. Semnalele de ieşire din *MIREG* (furnizate de circuite bistabile) sînt transmise la proces prin amplificatoare de adaptare la cablul de legătură. Ieşirile din *MIREG* se fac prin semnale numerice care pot comanda circuite integrate de tipul TTL;

— modulul de ieşiri prin trenuri de pulsuri *(MITI)* furnizează utilizatorului pulsuri de tensiune prin intermediul a 16 contacte cu relee închise pe o perioadă $\Delta t_p = 8 \div 250$ ms (după cum cere beneficiarul), care se pot succeda la intervale de timp $t_{k,k-1} = 16 \div 225$ ms (reglabile tot la cererea utilizatorului). Pentru generarea fiecărui puls, de pe o anumită ieşire, unitatea centrală *UC* (v. fig. 11.39) transmite cîte o instrucţiune de comandă a închiderii contactului releului. Comanda de deschidere a contactului releului este dată de către ansamblul de comandă al *DENA*. Pot exista maximum cinci astfel de module *MITI*:

— modulul de ieşiri pe relee cu contacte bistabile *(MIRA)*, relee care sînt cu automenţinere, poate furniza utilizatorului 1 la 56 grupe de cîte 16 contacte bistabile mecanice (deci maximum 896 relee). Memorarea informaţiei fiind făcută chiar prin contactele releelor, este suficient un singur registru de informaţie (cu 16 celule binare) pentru toate cele 56 grupe posibile, păstrarea informaţiei în acest registru fiind necesară numai pînă la „înscrierea" ei prin contactele releelor. Selecţia unui grup de relee şi a unui releu în grupă se face prin matrice cu relee (v. *Ex. 11—8*), folosindu-se semnalele *0G÷7G* (comanda selecţiei liniei) şi *C1÷7* (comanda selecţiei coloanei), produse de blocul pentru comanda releelor cu automenţinere (*CRA* — fig. 11.39). Nivelul semnalelor de comandă este de 30 V;

— modulul de ieşire analogică *(MIANA)* este format dintr-un registru cu 16 circuite bistabile şi un convertor numeric — analogic. Acest modul *(MIANA)* primeşte de la ansamblul de comandă: 16 semnale informaţionale; un semnal de validare a informaţiei: un semnal de linie *iL* ($i = 00,01,\ldots,63$) şi un semnal de iniţializare — *RZ*. Semnalele transmise procesului de acest modul sînt tensiuni: — 10 V÷ + 10 V şi 2÷ 10 V sau/şi cureţi: — 10 mA ÷ + 10 mA şi 2 ÷ 10 mA.

Ansamblul de comandă al *DENA* lucrează în modul următor: *UC* transmite informaţiile spre *DENA* în timpul unei instrucţiuni WD (cu *RWD* = „1"); prin apăsarea clapetei *INGDINA* (v. fig. 11.39) de la consola operatorului de proces *(COP)*, instrucţiunile transmise de *UC* devin inefective pentru *DENA*, iar modulul *MIREL* este iniţializat; celelalte module de ieşire ale *DENA* se iniţializează printr-un semnal produs prin apăsarea unei alte clapete *(INIT DINA)* de la consola operatorului de proces şi care se transmite la *DENA* prin intermediul *DIN*.

Simbolurile semnalelor din schema *DENA* (fig. 11.39) ne sînt cunoscute din prezentările celorlalte cuploare FELIX C-32 P (*DIN* — fig. 11.36 şi *DIA* — fig. 11.37), cu excepţia semnalului *Întrep.* (transmis de blocul pentru comanda releelor cu autoreţinere — *CRA* (v. fig. 11.39) spre *UC* prin intermediul *DIN*). Acest semnal reprezintă o cerere de întrerupere, lansată după terminarea ciclului de comutare a releelor cu contacte bistabile (din *MIRA*), pentru a anunţa *UC* că acest modul cu relee poate accepta o nouă instrucţiune WD.

Scanned with OKEN Scanner

Ex. 11—10. **Interfaţa standard IBM 1827**

Este o interfaţă complexă, care include atît interfaţa industrială (între nivelele 2 şi 4) cît şi unitatea de control pentru asigurarea legăturilor între un calculator universal şi interfaţa industrială (între nivelele 1 şi 2). IBM 1827 este şi o interfaţă hibridă care, în afara rolului de asigurare a conexiunilor cu instrumentaţia de proces (v. *Ex. 11—3*), poate fi utilizată şi la interconectarea în timp real a unui calculator numeric universal cu un calculator analogic în vederea realizării unui sistem de calcul automat hibrid.

În ceea ce ne priveşte, IBM 1827 poate asigura legătura simultană cu pînă la 288 puncte de intrare/ieşire numerice diverse (periferice standrad, procesoarele altor calculatoare — ca, de exemplu, IBM 360 sau IBM 370 — dintr-o reţea ierarhizată etc.), putînd fi — în acelaşi timp — şi interfaţă multiplă, calculator-proces, pentru calculatorul de proces IBM 1800 (v. fig. 11.8). În continuare vom face o prezentare a unităţii IBM 1827 privită numai ca interfaţă de proces, aşa cum este şi utilizată într-o mare întreprindere petrochimică din ţara noastră.

În această „postură" unitatea IBM 1827 asigură legătura calculatorului IBM 1800 cu procesul prin trei dispozitive de bază : pentru intrările analogice, pentru intrările numerice şi pentru ieşirile numerice şi analogice, ceea ce rezultă şi din schema generală a configuraţiei interfeţei IBM 1827 prezentată în figura 11.40.

Fluxul informaţional (al datelor) este reprezentat, într-un mod intuitiv, în schema din figura 11.41. Interfaţa IBM 1827 funcţionează în „data interleave mode" (v. fig. 11.9), deci poate fi conectată direct la canalul multiplexor al sistemului de calcul. Dispozitivelor periferice de proces, ataşate interfeţei, le sînt atribuite următoarele nivele de prioritate :

nivelul 0 — cereri de transfer de date iniţiate de canalul multiplexor al sistemului ;

nivelul 1 — cereri de transfer iniţiate de : 1.1 — intrările analogice, 1.2 — controlul convertoarelor analog-numerice, 1.3 — intrările numerice şi 1.4 — controlul convertoarelor numeric-analogice.

Datorită sistemului de priorităţi, orice transfer de date cu un periferic de proces poate fi temporar suspendat pentru a se executa transferul de date cu un periferic care are o prioritate mai înaltă.

Intrările analogice. Fluxul de date pentru aceste intrări este reprezentat schematic în figura 11.42.

Aşa-zisele „intrări" analogice prelucrează semnalele analogice prelevate din proces în vederea introducerii valorilor pe care le reprezintă în calculator. Pentru aceasta, în structura acestor intrări (reprezentată în figura 11.43) intră următoarele elemente de bază : convertoare analog-numerice, amplificatoare, comparatoare şi multiplexoare (*SC* din figura 11.43 reprezintă o sursă de comparaţie pentru măsurarea t.e.m., produse de termocuple, prin compensare).

Convertoarele analog-numerice lucrează cu semnale analogice sub formă de tensiuni electrice cuprinse în domeniul — 5 V la + 5 V. Principalele caracteristici tehnice ale convertoarelor sînt următoarele :

— rezoluţia selectabilă prin program (bits)	8	11	14
— durata conversiei (μs)	29	36	44

Scanned with OKEN Scanner

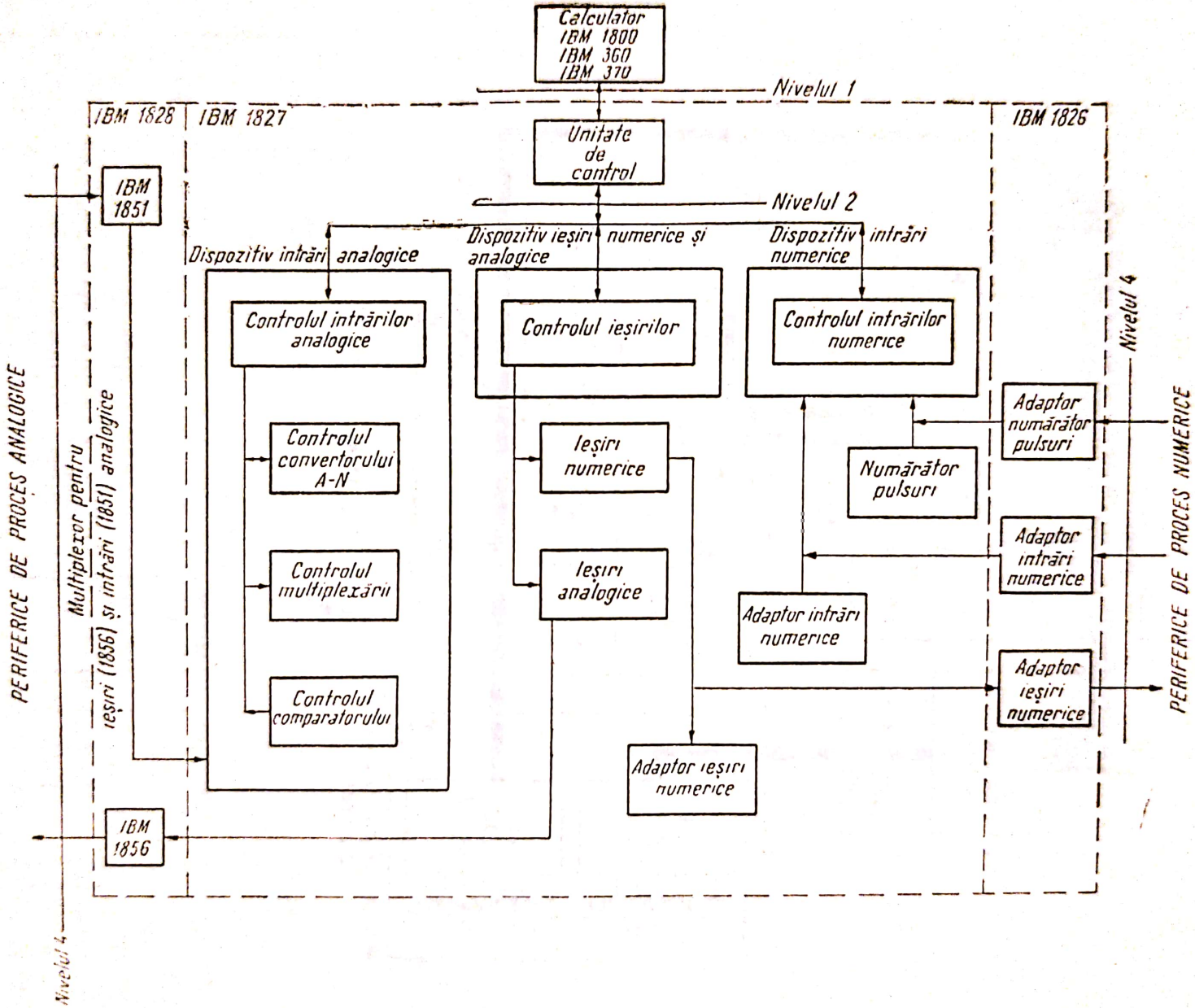


Fig. 11.40.

Scanned with OKEN Scanner

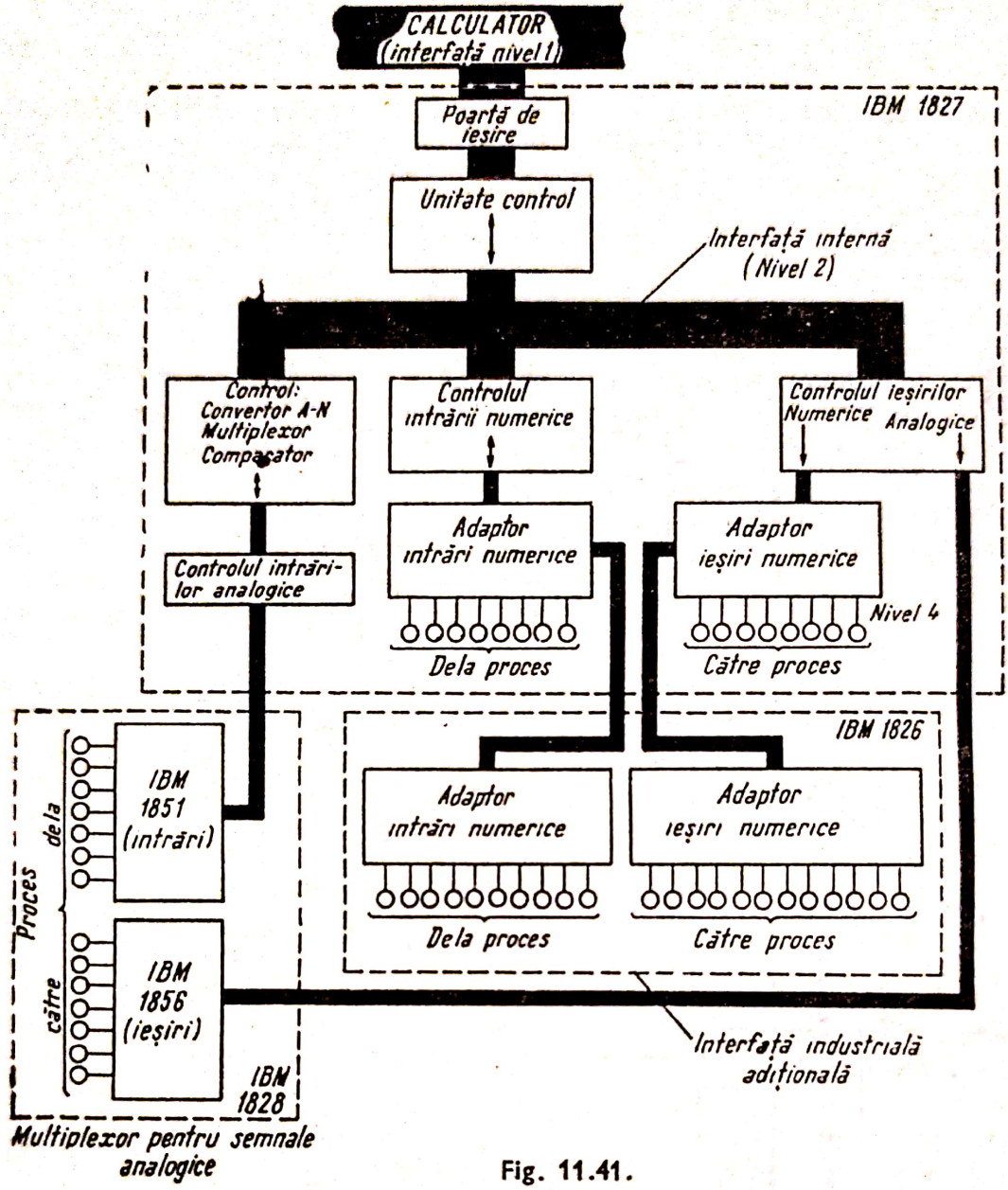


Fig. 11.41.

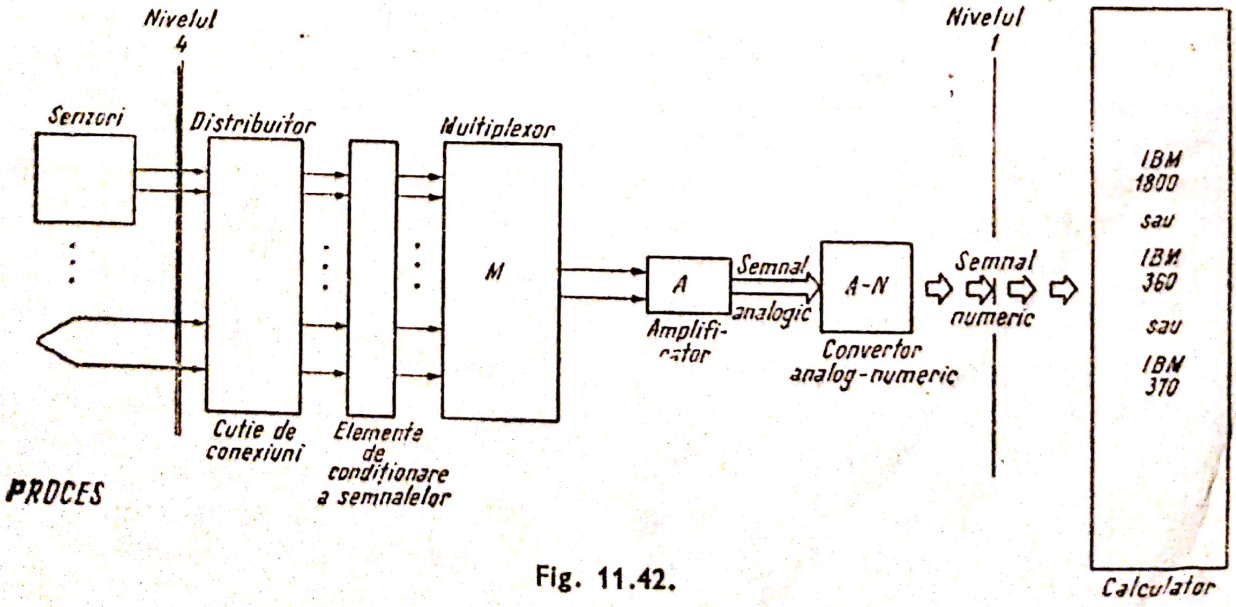


Fig. 11.42.

Scanned with OKEN Scanner

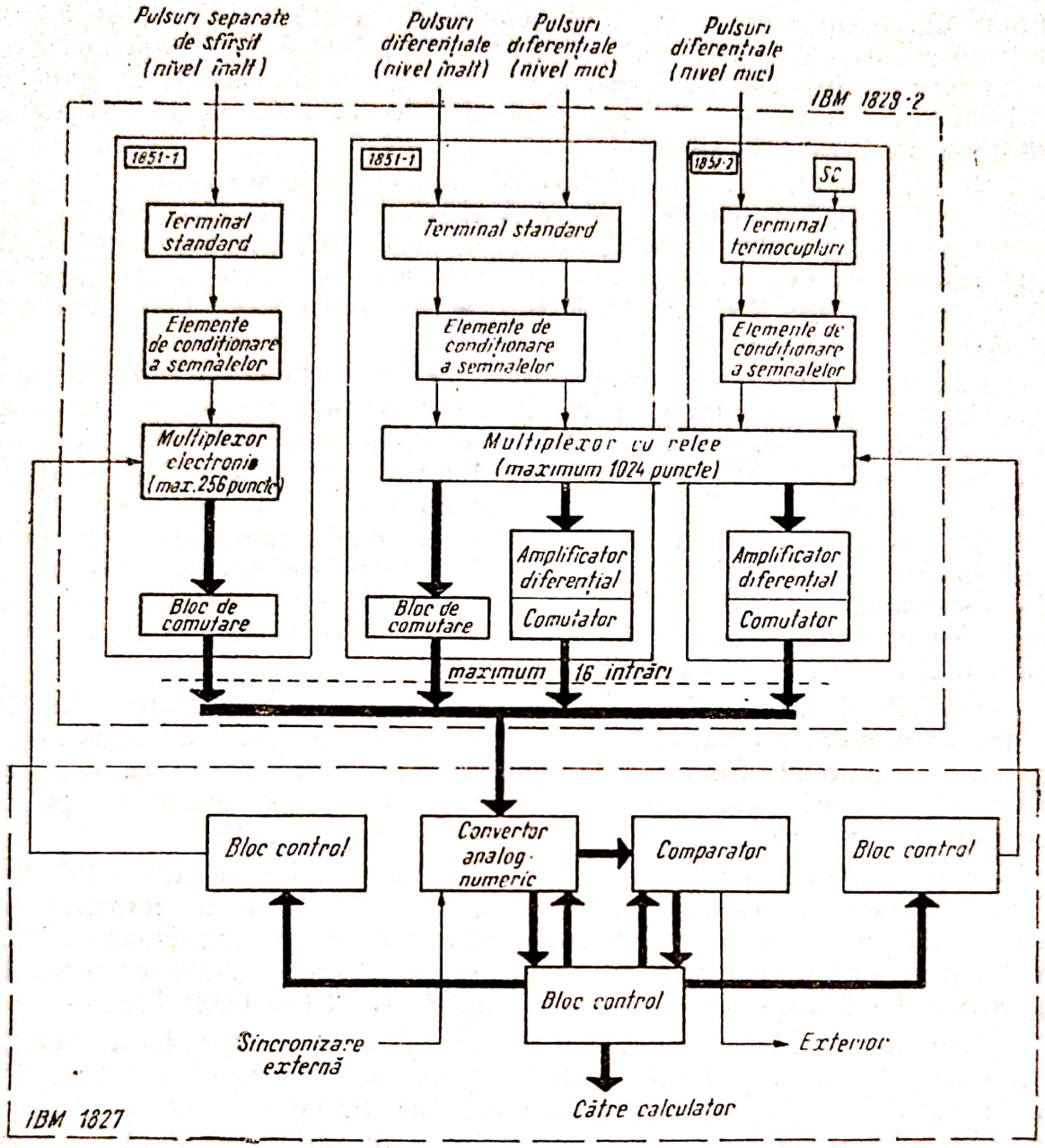


Fig. 11.43.

— frecvența (conversii/secundă) 35 100 28 000 23 000
— impedanța de intrare (MΩ) 10 10 10
— frecvența de transfer echivalentă pentru calculator 10 000 la 20 000 valori/secundă (include și operațiile de adresare, eșantionare și transmitere prin canale).

Amplificatoarele sînt incluse în convertoarele analog-numerice pentru a realiza nivelele de tensiune de 5 V necesare la intrarea în convertor. Amplificatoarele acestea pot lucra fie în regim continuu, fie în regim de eșantionare (cu reținere). Semnalele de intrare în aceste amplificatoare pot avea nivelele în gama 10÷500 mV.

Comparatoarele realizează verificarea automată a mărimilor analogice în raport cu limitele prescrise, care sînt transmise de calculator. Aceste

limite sînt comparate cu valorile de la ieşirea convertorului $A-N$ şi depuse într-un registru al convertorului. Dacă aceste valori sînt în afara limitelor, comparatorul iniţiază un transfer de date către canalul multiplexor al calculatorului indicîndu-i adresa punctului de măsurare din proces pentru care s-a produs depăşirea limitelor.

Multiplexoarele explorează intrările analogice şi stabilesc conectarea lor succesivă la convertoarele $A-N$. Multiplexoarele cu relee pot explora 64 grupe a 16 puncte (în total 1 024 puncte de intrare) cu o viteză de 100 puncte/secundă. Multiplexoarele electronice pot explora 16 grupe a 16 puncte (în total 256 puncte de intrare, însă cu o viteză foarte mare de 10 000 puncte/secundă).

Controlul intrărilor analogice dirijează funcţionarea convertoarelor $A-N$, a comparatoarelor şi a multiplexoarelor prin prelucrarea comenzilor de citire, de scriere şi de control iniţiate de calculator. Adresarea punctelor de intrare se poate face: secvenţial (printr-un singur subcanal multiplexor al calculatorului) şi aleator (prin două subcanale multiplexoare ale calculatorului). Verificarea limitelor (prin comanda funcţionării comparatoarelor) impune folosirea a trei subcanale multiplexoare ale calculatorului: două subcanale pentru verificarea limitelor prin adresare aleatoare şi un al treilea pentru transmiterea de către comparator calculatorului a adresei punctului de intrare la care s-a produs ieşirea din limite.

Intrările numerice. Ele fac posibilă recepţionarea de către calculator în timp real a datelor numerice, transmise din proces prin semnale numerice: pulsuri independente (produse de contacte sau salturi/nivele de tensiune), şiruri de pulsuri care reprezintă numere binare şi grupe de pulsuri exprimînd cifre zecimale prin codificare binară (cod BCD).

Structura unui cuplor numeric al interferenţei standard IBM 1827 corespunde schemei de principiu din figura 11.34. O intrare numerică are 8 adaptoare numerice cu cîte 8 grupe numerice de intrare; deoarece informaţia numerică este organizată în grupe de 16 bits, rezultă că o intrare numerică IBM 1827 are capacitatea de $8 \cdot 8 \cdot 16 = 1\,024$ bits.

Comanda intrărilor numerice se face prin instrucţiunile de scriere/citire directă şi prin modulul „controlul intrărilor numerice" (notat cu *CIN* în figura 11.34). Prin utilizarea unor comenzi de citire modificate se pot include: operaţii de sincronizare externă, adresări secvenţiale, adresări individuale sau forţarea citirii* în „burst mode" (v. pag. 141). O comandă de control de la calculator (IBM 1800, IBM 360 sau IBM 370) determină transferul a doi octeţi („Byte") care sînt încărcaţi în registrul de adresare al „controlului intrărilor numerice", ca adresă de strat. După recepţionarea adresei de start, o comandă de citire a calculatorului iniţiează execuţia transferului de date din grupul de intrare adresat către registrele de date ale „controlului intrărilor numerice" (16 bits). Apoi, „controlul intrărilor numerice" cere transferul către calculator şi depune datele în memoria lui internă. „Controlul intrărilor numerice" (*CIN* — fig. 11.34) termină operaţia de explorare a intrărilor atunci cînd primeşte ca răspuns la cererea de transfer a datelor o comandă de stop. În această situaţie, „contro-

---

* „Forţarea" citirii sau scrierii se referă la registrele tampon şi constă în preluarea datelor sau încărcarea registrului cu o nouă valoare fără trecerea prin zero.

Scanned with OKEN Scanner

lul intrărilor numerice" semnalează sfîrşitul transferului de date şi readuce registrele la zero.

Celelalte module ale unei intrări numerice au fost prezentate cu ocazia comentării figurii 11.34.

Ieşirile numerice şi analogice. Configuraţia acestui cuplor al interfeţei IBM 1827 este reprezentată în figura 11.44 din care rezultă că pentru „ieşirile" spre proces sînt prevăzute două secţiuni distincte: ieşirile numerice şi ieşirile analogice.

Ieşirile numerice sînt organizate în trei compartimente: porţi electronice („Electronic Contact Operate" — *ECO* pe figura 11.44); ieşiri pentru pulsuri („Pulse Output" — *PO* pe figura 11.44) şi registre de ieşire („Register Output" — *RO* pe figura 11.44).

Porţile electronice *(ECO)* sînt utilizate pentru a comanda dispozitive de acţionare din proces (executori) ca: relee, servomotoare de curent continuu etc., precum şi aparatură de indicare ca: lămpi de tablou, ecrane cu afişaj numeric, sonerii de alarmă etc. Porţile electronice de ieşire sînt aranjate în grupe de cîte 16, putînd fi cel mult 108 grupe (deci, în total,

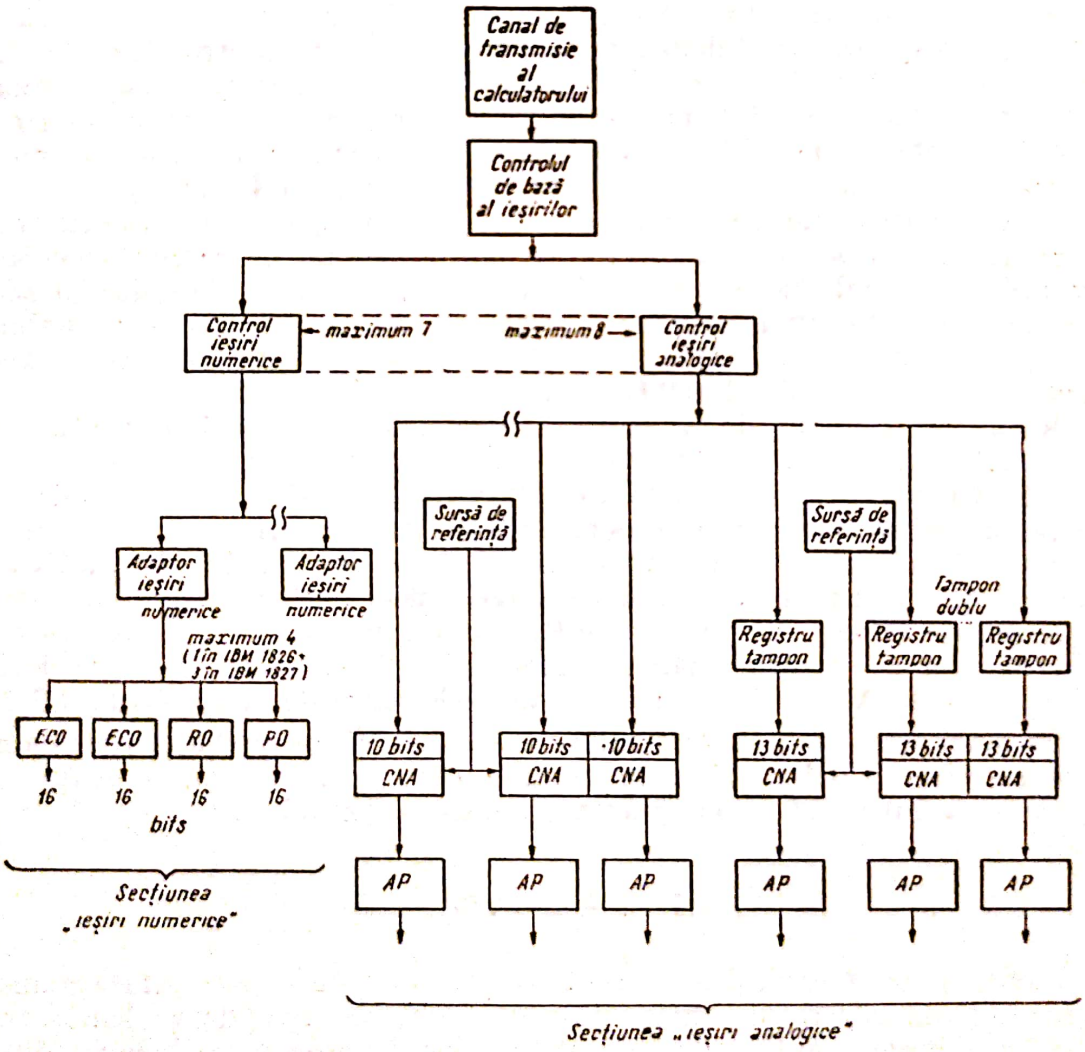


Fig. 11.44.

pot fi 1 728 puncte de ieşire numerice). Cele 16 porţi ale unei grupe sînt comandate de ieşirea unui registru cu 16 bits şi rămîn condiţionate pînă cînd registrul primeşte un nou set de date. Bitul „1" corespunde porţii închise (conducţie), iar bitul „0" corespunde porţii deschise (circuit întrerupt).

Ieşirile pentru pulsuri *(PO)* sînt folosite pentru a comanda, prin trenuri de pulsuri, dispozitivele specifice ca: circuite basculante bistabile, porţi, servomotoare pas cu pas etc. *PO* sînt aranjate în grupe de cîte 16, numărul maxim de grupe putînd fi 108 (deci, în total, 1 728 puncte). Cele 16 ieşiri *PO* ale unei grupe sînt comandate de ieşirea unui registru cu 16 bits. Cele 16 puncte de ieşire sînt condiţionate la transferul de date de la calculator (încărcarea registrului) şi rămîn active pînă la comanda de readucere la zero a registrului (dată de un „timer" separat) cu perioada de 3 ms. Un bit „1" corespunde ieşirii active (pulsurile trec spre ieşire), iar un bit „0" corespunde ieşirii inactive (pulsurile sînt blocate).

Registrele de ieşire *(RO)* asigură transferul de date sub formă numerică între calculator şi periferice (standard sau speciale de proces). Un registru de ieşire are maximum 108 grupe a cîte 16 bits (în total 1 728 bits). Fiecare grupă de 16 bits poate fi transmisă din registrul de ieşire către registrul corespunzător al dispozitivului periferic. Sub controlul sistemului, datele din registrul de ieşire pot fi scrise repetat în registrul perifericului.

Ieşirile analogice sînt realizate cu ajutorul convertoarelor numeric-analogice (notate cu *CNA* pe figura 11.44), prevăzute cu registre tampon la intrare şi cu amplificatoare de putere (*AP* — fig. 11.44) la ieşire.

Convertoarele numeric-analogice (*CNA*) au următoarele caracteristici principale: rezoluţia la intrare 13 bits şi un bit de semn; domeniul tensiunii electrice de la ieşire — 5 V la + 5V; impedanţa la ieşire 10 kΩ; precizia $\pm 10^{-4}$; stabilitatea (pe durata de 40 ore) $\pm 5 \cdot 10^{-5}$; stabilitatea (în regim de durată: 6 luni) $\pm 10^{-4}$; timpul de conversie 10μs; frecvenţa de conversie 100 000 conversii/secundă.

Registrele tampon de intrare ale convertoarelor *CNA* sînt utilizate pentru mărirea vitezei de transfer a datelor între canalul calculatorului şi *CNA*. Un astfel de registru tampon („buffer") are 64 de secţiuni adresabile, permiţînd să se realizeze o ieşire analogică prin *CNA* pentru 64 puncte numerice. Caracteristic pentru aceste „baffer" -e este modul de încărcare „forţată" (adică cu introducerea unei noi valori fără trecere prin zero).

Amplificatoarele de putere *(AP)* extind posibilităţile de utilizare a convertoarelor *CNA* prin mărirea domeniului tensiunilor de ieşire de la ± 5 V la ± 10 V şi reducerea impedanţei de ieşire de la 10 kΩ la 0,6 Ω.

Să observăm, ca o calitate importantă, gradul de modularitate înaintat al interfeţei IBM 1827, care permite realizarea oricăror aplicaţii cu minimum de investiţii şi asigurarea treptată a extinderilor.

11.4. APLICAŢII

Încheiem acest capitol prin trei exemple concrete — prezentate numai din punctul de vedere al echipamentului — cu intenţia de a demonstra diversitatea mare a aplicaţiilor calculatoarelor de proces, în domenii diferite şi cu structuri de sistem specializate, însă obţinute cu module de bază.

Scanned with OKEN Scanner

Ex. 11—11. **Tehnica analizelor**

Scopul principal al automatizării operaţiilor de analiză a substanţelor (produse finale sau/şi intermediare într-un proces tehnologic) este acela al scurtării timpului de prelucrare a rezultatelor analizelor şi efectuarea lor în timp real (aici cu înţelesul de timp util). Folosind un calculator de proces, acest deziderat se obţine prin: prelevarea datelor de la analizoare, înregistrarea lor în memoria internă a calculatorului de proces, prelucrarea datelor conform unor programe de aplicaţii catalogate în sistemul de operare şi indicarea operativă (la „ieşiri") a rezultatelor.

Această tehnică de analiză cu ajutorul calculatorului de proces se poate face în două moduri principial diferite: unul constă în conectarea la interfaţa sistemului a mai multor analizoare de acelaşi tip (de exemplu: mai multe cromatografe de gaze), iar altul se referă la utilizarea „intercalată" a unor aparate de analiză de tipuri diferite (de exemplu: spectrometre în ultraroşu, spectrometre de rezonanţă nucleară, spectrometre de masă, cromatografe de gaze ş.a.).

În cazul analizelor combinate, de exemplu a cromatografiei de gaze şi spectrometriei de masă (v. fig. 11.46), calculatorului de proces îi sînt specifice modalităţile de înregistrare a datelor măsurate (prin aparatele de analiză) şi procedurile de prelucrare a lor conform tehnicii analizelor. În cadrul acestui exemplu, ne vom limita numai la modalitatea de prelevare şi interpretare a datelor printr-un cromatograf de gaze şi la prezentarea configuraţiei calculatorului de proces [0].

Gazul de analizat, condus prin coloana de separaţie a unui cromatograf, după ce a fost descompus în componentele sale — prin mai mulţi timpi de parcurgere — ajunge într-un detector (un senzor de tip traductor electric). Detectorul produce la ieşirea sa o tensiune electrică proporţională cu concentraţia fiecărui component (un semnal analogic). Reprezentarea grafică a variaţiei în timp a acestei tensiuni electrice este cunoscută sub numele de cromatogramă (un exemplu de cromatogramă este dat în figura 11.45). Timpul scurs de la începutul analizei pînă a apariţia maximului fiecărui component (notat cu t_R pe figura 11.45 şi numit timp de „retenţie") reprezintă o indicaţie privind felul componentului. Suprafaţa SV corespunzătoare unui „vîrf" al cromatogramei (v. fig. 11.45) reprezintă o măsură a concentraţiei acelui element component care a produs „vîrful".

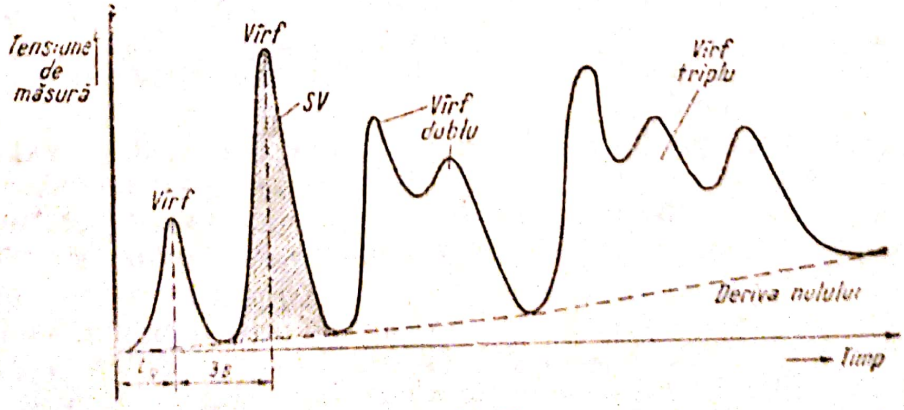


Fig. 11.45.

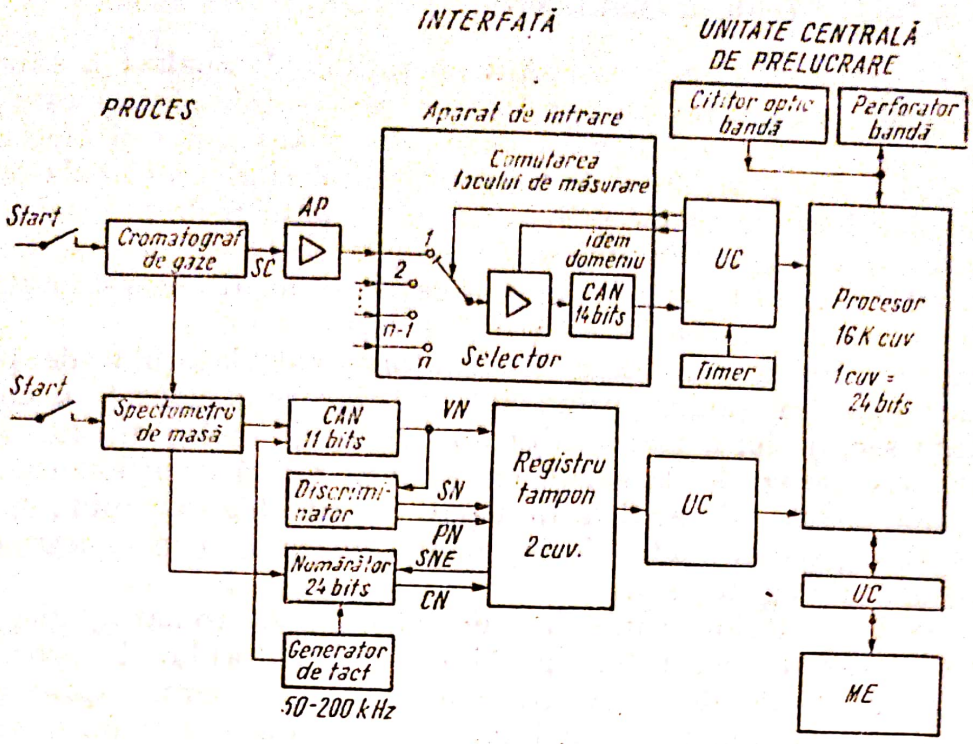


Fig. 11.46.

Deoarece cîteodată, cîțiva dintre componenți sînt foarte „apropiați" unul de altul, apar și vîrfuri multiple (duble, triple etc.).

Semnalele analogice de tensiune ale cromatografiei sînt transmise calculatorului de proces. O configurație posibilă pentru un calculator de proces destinat analizelor este prezentată în schema din figura 11.46, unde *SC* reprezintă semnalul analogic (tensiuni electrice) produs de cromatograful de gaze și introdus în interfața industrială printr-un amplificator primar *AP*. Amplificatorul primar este comutabil, cu comandă prin program, astfel încît poate lucra cu 4 game de semnale (cu valori extreme pentru *SC* 1μV și 1 V). Celelalte notații din figura 11.46 au următoarele semnificații: *CAN* — convertoare analog-numerice, *UC* — unități de control, *ME* — unități de memorie externă standard, *VN* — semnale cu valori numerice, *SN* — semnal pentru validarea preluării valorii numerice, *PN* — comanda preluării valorii numerice; *SNE* — șir de numere de comparare; *CN* — conținutul numărătorului. Abrevierea *cuv.* reprezintă „cuvîntul", adică unitatea informațională de prelucrare (cu lungimea, aici, de 24 bits).

Deoarece distanța dintre vîrfuri este în medie de 3 secunde (v. fig. 11.45), iar pentru determinarea suprafeței unui vîrf prin metoda numerice sînt necesare cca 20÷30 intervale (puncte de discretizare), rezultă că în acest caz rata de eșantionare a convertoarelor analog-numerice este relativ scăzută.

Fiecare valoare măsurată este prelucrată prin program, la intervale de tact de 1/10 secunde (uneori și 1/200 s). Calculatorul determină timpul de retenție t_R (din durata de apariție a volorilor maxime față de momentul inițial) și suprafața *SV* (v. fig. 11.45) prin metode de integrare nume-

Scanned with OKEN Scanner

rice. Apoi, comparînd valorile t_R şi SV cu constantele dintr-o listă memorată iniţial, indică elementul component şi concentraţia lui.

Apar şi cîteva probleme mai dificile ca : suprapunerea vîrfurilor („aglomerarea" suprafeţelor de vîrf) şi deriva nulului (v. fig. 11.45) introdusă de coloana cromatografului. Prima problemă se rezolvă prin mărirea ratei de eşantionare şi extrapolări (refacerea alurii curbei pentru fiecare vîrf dintr-un grup multiplu), iar a doua prin corecţii realizate de programul de calcul.

Dacă este necesar să se determine şi diferiţii izotopi ai unui component, atunci se cuplează la calculatorul de proces şi un spectrometru de masă.

Încheiem acest exemplu cu observaţia că pînă în prezent s-au şi realizat, în ţara noastră, analize industriale prin utilizarea calculatorului de proces românesc FELIX C-32P.

Ex. 11—12. **Analiza comportării în exploatare a zăcămintelor petrolifere**

După cum se ştie, zăcămîntul petrolifer reprezintă un complex termohidrogazodinamic unitar, constituit din cele două zone (productivă şi acviferă) neseparate prin limite etanşe, deci cu fenomene de interferenţă, cu o structură eterogenă atît în privinţa proprietăţilor fluidelor conţinute cît şi a rocii colectoare, precum şi ale sistemului fluid-rocă. În aceste condiţii, comportarea în exploatare a unui zăcămînt petrolifer depinde de factori naturali determinaţi de caracteristicile geologice, fizice şi hidrogazodinamice ale zăcămîntului. Numărul mare şi diversitatea acestor factori, precum şi modificarea — pe măsură ce procesul de exploatare evoluează — a informaţiei cu privire la aceşti factori şi a imaginii zăcămîntului, fac ca soluţiile optime ale problemelor de exploatare să difere de la o etapă la alta. Din această cauză, pentru realizarea permanentă a unui proces optim de exploatare, sînt necesare numeroase studii de analiză, calcule de prevedere a comportării zăcămîntului şi de proiectare a exploatării, efectuate periodic şi în mai multe variante posibile (în prezent, s-a ajuns la necesitatea ca pentru fiecare obiectiv în parte proiectarea exploatării lui să fie refăcută la intervale relativ scurte, chiar şi de cîte 6 luni !).

În aceste condiţii şi exigenţe (multe obiective, volum mare de date, numeroase variante de executat şi interpretat, corelaţia între stările evolutive ale zăcămîntului, durata mică a studiului, precizia ridicată cerută soluţiilor ş.a), singura tehnică în stare să realizeze conducerea optimală a exploatării zăcămintelor petrolifere este **simularea hibridă***, care reprezintă o aplicaţie particulară a calculatoarelor de proces.

Ea constă în „înlocuirea" zăcămîntului — ca sistem original — printr-un simulator format dintr-o reţea electrică plană, alcătuită din rezistoare R şi condensatoare C conectate în diverse moduri, cel mai adesea în forma unei reţele cu ochiuri patratice (fig. 11.47). În acest fel, sistemul original (zăcămîntul) devine o instalaţie electrică, iar procesele naturale de scurgere a petrolului cu o anumită repartiţie a presiunilor de zăcămînt

---

* Pentru detalii, recomandăm consultarea referatului „*Simularea sistemelor fizice cu ajutorul calculatoarelor cu structură hibridă*" de I. Dumitrescu, apărut în revista „Studii şi cercetări de mecanică aplicată" nr. 4, 1977, tomul 36.

Scanned with OKEN Scanner

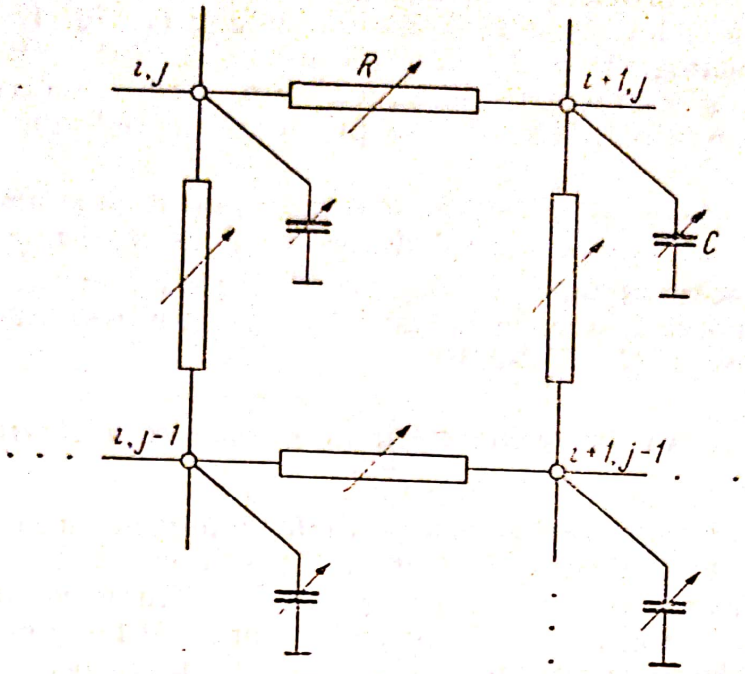


Fig. 11.47.

şi a debitelor (de petrol extras sau de fluid injectat) sînt înlocuite prin procese de laborator referitoare la repartiţia tensiunilor la nodurile i, j ale unei reţele electrice, respectiv cu curenţilor electrici din laturile de alimentare a reţelei în noduri.

De fapt, procesele de scurgere staţionară a hidrocarburilor fluide printr-un mediu poros sînt descrise, formal, de acelaşi model matematic ca şi procesele electrice staţionare dintr-un mediu conductor, adică prin ecuaţii cu derivate parţiale de tip parabolic care formează probleme cu condiţii la limită (pe contur) şi iniţiale. Mai exact, o reţea RC, de forma celei din figura 11.47, reprezintă o aproximare discretă a zăcămîntului continuu, iar modelul devine un sistem de ecuaţii algebrice liniare (cu necunoscute: potenţialele electrice ale nodurilor $V_{i,j}$ $i, j \in N$) care constituie o aproximare cu diferenţe finite a ecuaţiilor parabolice. După cum se ştie, în această tehnică R reprezintă (la o anumită scară) parametrii: ρ — densitatea fluidelor, k — permeabilitatea locală a mediului şi μ — viscozitatea fluidului, iar C simulează pe m — porozitatea locală a mediului şi β — coeficientul de compresibilitate.

Astfel, procesul de analizat constă în procesul de repartiţie a tensiunilor la noduri şi a curenţilor în laturi. Deoarece, pentru o bună aproximaţie, trebuie utilizate reţele cu un număr mare de noduri şi cu parametri $R-C$ reglabili în timp real, determinarea (măsurarea) potenţialelor la noduri, scrierea lor, trasarea unor grafice cu linii echipotenţiale (corespunzătoare izobarelor din zăcămînt) şi modificarea parametrilor $R-C$ pentru analiza mai multor variante, nu se poate face decît numai cu ajutorul unui calculator automat şi a unei interfeţe hibride (simulator — calculator). Toate acestea: simulator de tip reţea, interfaţa hibridă şi calculatorul automat formează un sistem tipic de genul calculatorului de proces.

Scanned with OKEN Scanner

Un exemplu de un astfel de calculator de proces, destinat studiului proceselor de filtraţie prin intermediul unui simulator analogic este reprezentat prin schema de principiu din figura 11.48. Acest calculator de proces este format din:

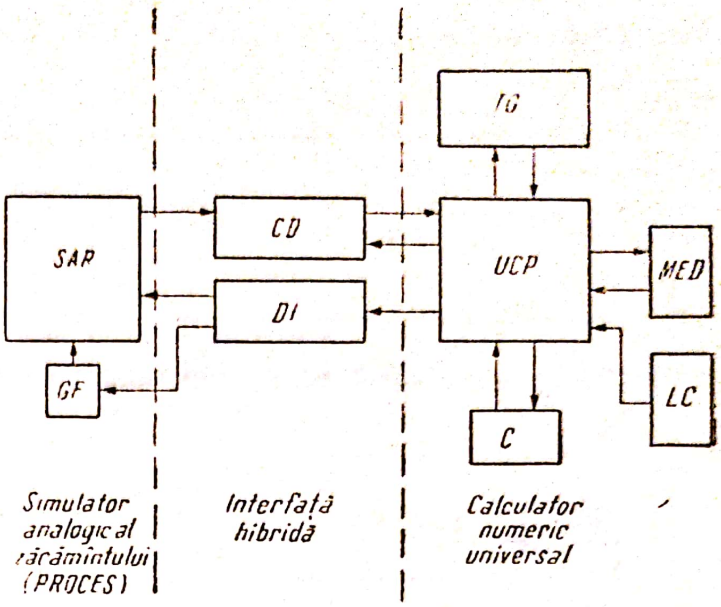


Fig. 11.48.

— *simulatorul analogic al zăcămîntului* (ca parte a procesului) alcătuită dintr-o reţea electrică $R - C$ plană (notată cu SAR în figura 11.48) şi un generator de funcţii *(GF)* care asigură condiţiile pe frontieră şi la sondele simulate prin valori reglabile de tensiuni electrice. Reţeaua de simulare a zăcămîntului este bine să aibă un număr suficient de noduri (i, j — v. fig. 11.47) pentru o cît mai bună discretizare (astfel, un număr de 1 280 noduri este considerat ca satisfăcător). Pentru o cît mai fidelă redare a zăcămîntului, este necesar ca rezistoarele din laturi şi condensatoarele nod-masă să fie reglabile. Se constată a fi suficient un reglaj în 10 trepte; aceasta se poate realiza prin: rezistoare cuplate în serie în laturile reţelei, condensatoare conectate în paralel între fiecare nod al reţelei şi masă, la care se adaugă relee electromagnetice cu contacte de şuntare a rezistoarelor şi de conectare a condensatoarelor, aşa cum se arată în schema din figura 11.49, unde este redat un singur nod (i, j) şi o singură latură (între nodurile i, j şi $i + 1, j$). Aici: $R_1, R_2, \ldots, R_{10}$ sînt cele zece trepte ale rezistorului din latură; $C_1, C_2, \ldots, C_{10}$ — cele zece trepte ale condensatorului de la noduri: $k_1, k_2, \ldots, k_{10}$ — contactele de şuntare a rezistoarelor, comandate de bobinele respective $K_1, K_2, \ldots, K_{10}$; $l_1, l_2, \ldots, l_{10}$ — contacte de cuplare a condensatoarelor la un nod, comandate de bobinele respective $L_1, L_2, \ldots, L_{10}$. Măsurarea potenţialelor nodurilor se face cu ajutorul unor selectoare cu contacte sau electronice;

— *calculator numeric universal*, care realizează următoarele faze ale simulării:

introducerea datelor printr-un lector de cartele perforate (*LC* în fig. 11.48). Datele introduse se referă la: forma conturului, parametrii ρ, k şi μ pentru fiecare latură (prin care se determină valorile rezistenţelor reţelei), parametrii m şi β pentru fiecare nod (prin care se determină capacităţile condensatoarelor), mărimile de lucru — presiuni şi debite pe contur şi în anumite noduri (transpuse în tensiuni şi curenţi electrici de generatorul de funcţii *GF*), amplasarea sondelor şi frontului de dislocuire, toate în mai multe variante posibile;

prelucrarea datelor (citite prin *LC* sau/şi prelevate prin măsurători pe simulatorul *SAR*) în vederea transpunerii pe reţea sau a corectării operative a parametrilor reţelei în funcţie de măsurătorile de presiune, precum

Scanned with OKEN Scanner

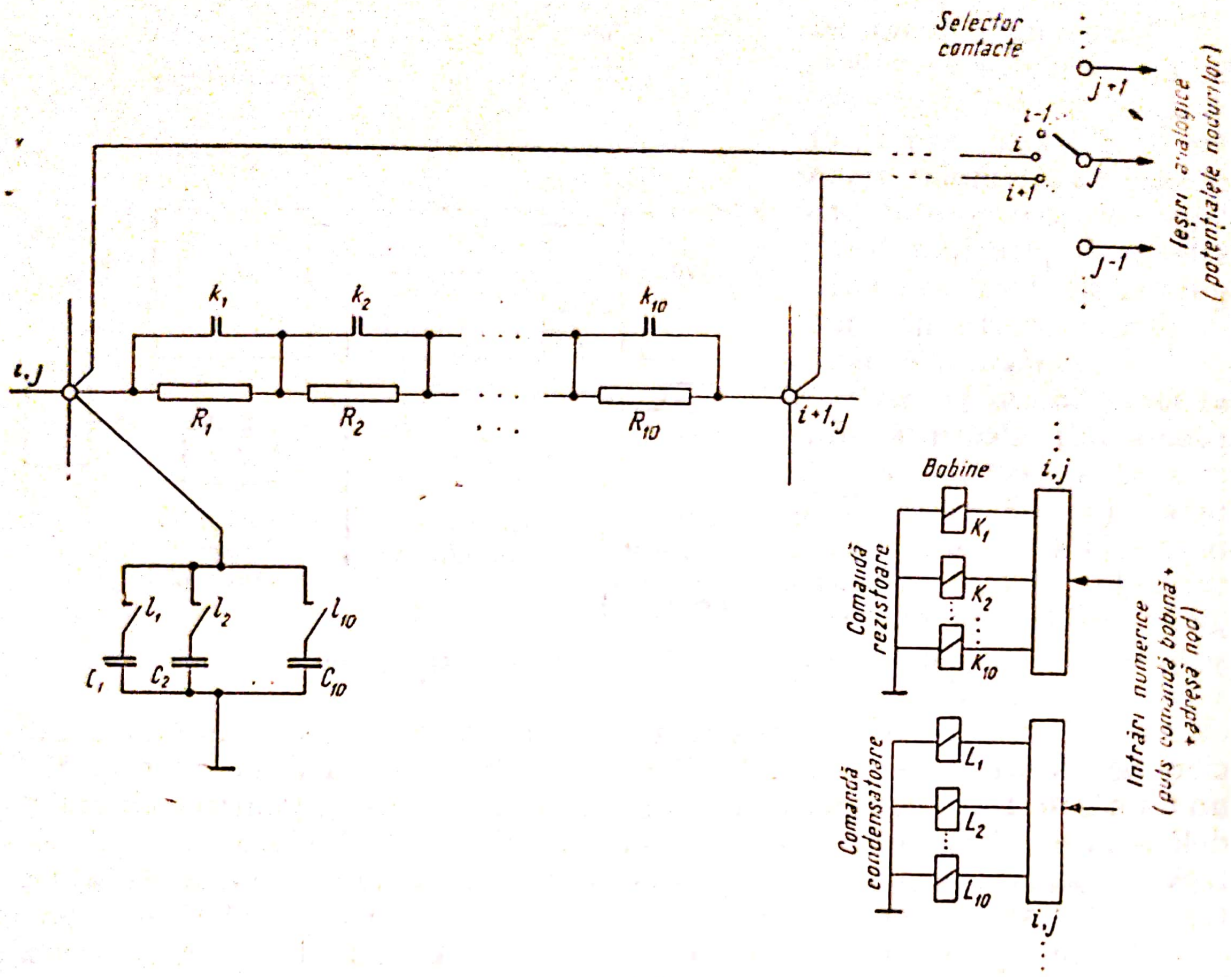


Fig. 11.49.

şi interpretarea finală a rezultatelor în vederea alegerii variantei optime. În acest scop se foloseşte o unitate centrală de prelucrare *(UCP)* de tip „clasic" (universal) a cărei memorie internă este suficient să aibă capacitatea de 32 Kocteţi;

memorarea datelor privind evoluţia şi istoricul zăcămîntului, precum şi a pachetelor cu programe de aplicaţie, cu ajutorul unei unităţi periferice cu discuri magnetice *(MED)*;

extragerea soluţiilor sub forma unor grafice realizate de un trasor de grafice *(TG)*, ca de pildă — un „digital plotter";

efectuarea unor corecturi de către un operator uman sau comunicarea de mesaje printr-o consolă *(C)*;

— *interfaţa hibridă*, cu totul specifică acestei aplicaţii, este compusă dintr-un aşa-numit colector de date *(CD)* şi un „distribuitor de informaţie" *(DI)*. Colectorul de date este format dintr-un canal multiplexor (de exemplu cu 128 subcanale) care explorează reţeaua simulatorului *SAR* nod cu nod, pe grupe de 128 noduri, şi un convertor analog-numeric. Distribuitorul de informaţie *(DI)* se compune dintr-un decodificator de adresare care comandă releele *K* (de şuntare a rezistoarelor) şi *L* (de conectare a condensatoarelor), relee ce reprezintă executorii procesului, conectaţi la ieşirile numerice ale interfeţei hibride. Distribuitorul de informaţie comandă şi generatorul de funcţii *GF* în scopul realizării condiţiilor la limită (pe contur şi în locurile de amplasare a sondelor).

Scanned with OKEN Scanner

Ex. 11—13. **Sistem de conducere pentru liniile de prelucrări mecanice**

În figura 11.50 reproducem, din [29], schema de principiu a unui sistem de conducere centralizat dintr-o întreprindere constructoare de mașini. Fiecare linie tehnologică a întreprinderii, ca de exemplu o linie de prelucrări mecanice *PM*, este condusă automat de un calculator propriu de proces *CP* (în figura 11.50 un calculator IBM System/7). Acest calculator de proces poate fi conectat și cu un calculator pentru evidență *CPE* (de exemplu, un „desk computer" IBM 1130), care realizează prelucrări de date privind evidența producției, gestiunea materialelor și situația stocurilor, sau cu un calculator universal central de mare putere *CU*

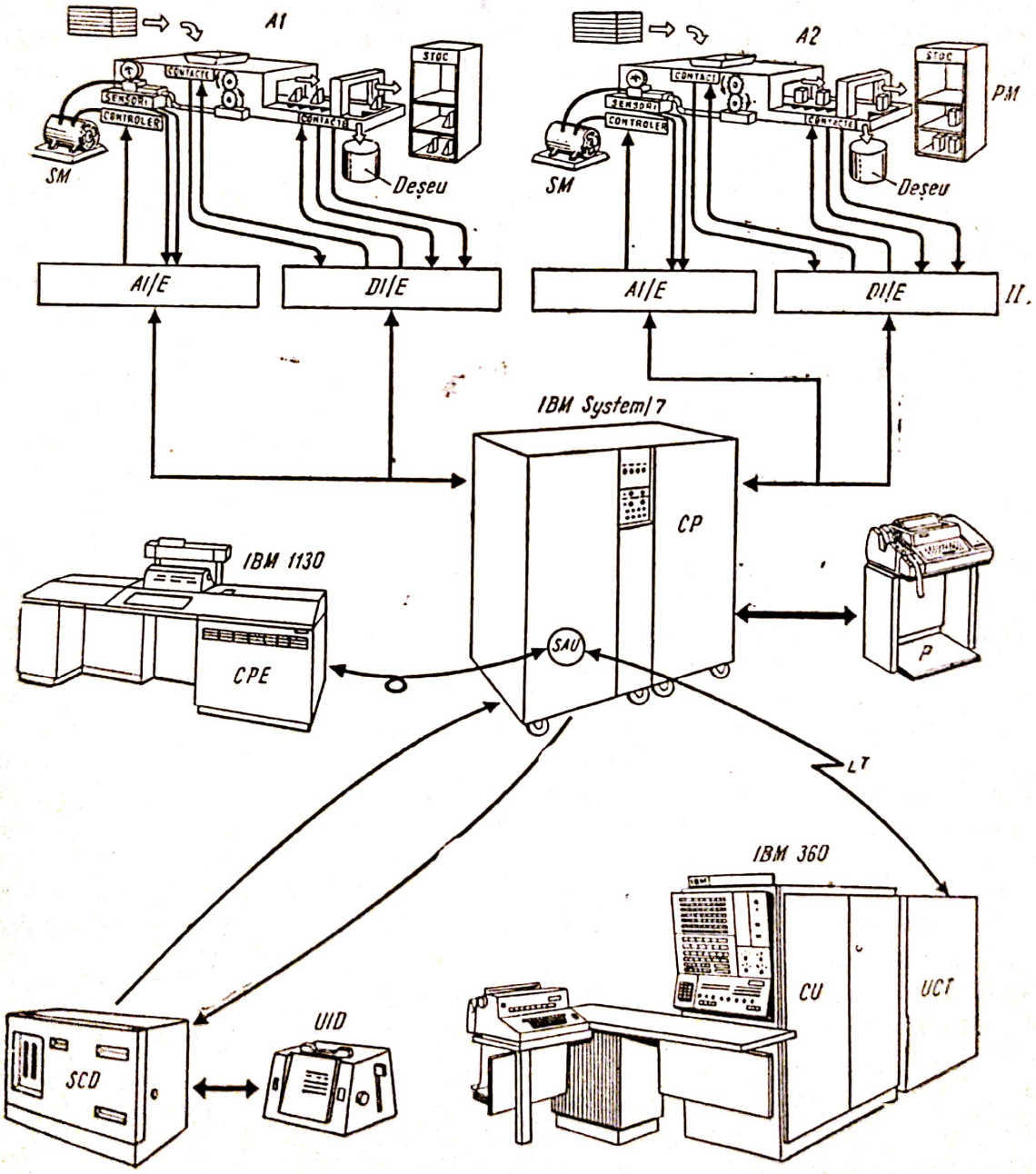


Fig. 11.50.

Scanned with OKEN Scanner

(de exemplu un sistem IBM 360 sau IBM 370), printr-o linie de telecomunicații *LT* și o unitate de adaptare și control (de tipul „start/stop") al teleprelucrării *UCT*.

Conectarea calculatorului de proces IBM System/7 la linia tehnologică se face printr-o interfață industrială *II* (formată în principal din „intrări/ieșiri" pentru semnalele analogice *AI/E* și „intrări/ieșiri" pentru semnalele digitale *DI/E*). Instrumentația de proces, montată la agregatele tehnologice *A1*, *A2*, ... este formată din: senzori de turație, numărătoare de piese, senzori pentru parametrii procesului de prelucrare prin așchiere, contacte etc. și executori (ca servomotoare *SM*, controlere, relee de blocare etc.).

La calculatorul de proces *CP* mai sînt conectate unitatea periferică proprie de intrare/ieșire (*P*) și un sistem de colectare și comunicare a datelor (*SCD*) privind evidența timpului lucrat de angajați, evidență personal, evidență ieșiri și intrări din magaziile de materiale, scule, piese de schimb materii prime etc. (realizată prin unități de introducere a datelor de evidență *UID*).

Un sistem complex, de forma celui din figura 11.50, poate realiza o conducere ierarhizată (la nivele ca: linii tehnologice/agregate, secții, uzine, întreprinderi și centrală industrială) asigurînd implementarea unui sistem informatic unitar și eficient.

Scanned with OKEN Scanner

Capitolul 12

SISTEME DE PROGRAME

După cum se ştie [1, 12], totalitatea programelor care asigură conducerea procesului de prelucrare a datelor, planificarea resurselor sistemului de calcul, supravegherea şi controlul lucrărilor („job-''urilor) — dirijîndu-le spre diferite echipamente ale sistemului, în vederea executării programelor de aplicaţie (ale utilizatorilor) — alcătuiesc ceea ce se numeşte un *sistem de programe.*

Un sistem de calcul (sistem de prelucrare automată a informaţiei) nu poate funcţiona fără existenţa unui sistem de programe, adică un set de instrucţiuni ale calculatorului, memorate în sistemul de calcul, pentru a realiza succesiunea operaţiilor în vederea obţinerii rezultatului în conformitate cu algoritmul ales.

Există diferite modalităţi de clasificare şi grupare a sistemelor de programe. În lucrarea de faţă, acestea vor fi clasificate în sisteme de operare şi programe de aplicaţie (ale utilizatorilor).

Programele de aplicaţie sînt acele programe prin a căror execuţie se rezolvă o problemă concretă pentru care a fost destinat calculatorul de proces. Exemple de asemenea probleme sînt: supravegherea, protocolarea şi bilanţarea automată a unui proces chimic, reglarea numerică directă a unui proces sau conducerea automată a unei linii automate de motoare etc.

Programele care alcătuiesc sistemul de operare sînt, în principiu, independente de aplicaţiile concrete la care este utilizat calculatorul de proces. Acestea intervin pe parcursul diferitelor etape de execuţie ale unui program aplicativ, supraveghind desfăşurarea acesteia, precum şi funcţionarea fără erori a sistemului de calcul. Sistemul de operare asigură o comunicaţie eficientă între operator şi calculator prin intermediul mesajelor care marchează diferitele etape de execuţie ale programelor aplicative.

În afara celor două categorii de programe, există o altă categorie de programe standard ce sînt construite în forma specifică fiecărui calculator şi stau la dispoziţia utilizatorului în aşa-numita bibliotecă de programe.

Referindu-ne la calculatoarele moderne, trebuie să precizăm că în prezent un sistem automat de prelucrare automată a datelor este privit ca fiind reunirea funcţională a trei părţi distincte:

— sistemele de echipamente, adică partea numită şi hardware (v. cap. 11);

Scanned with OKEN Scanner

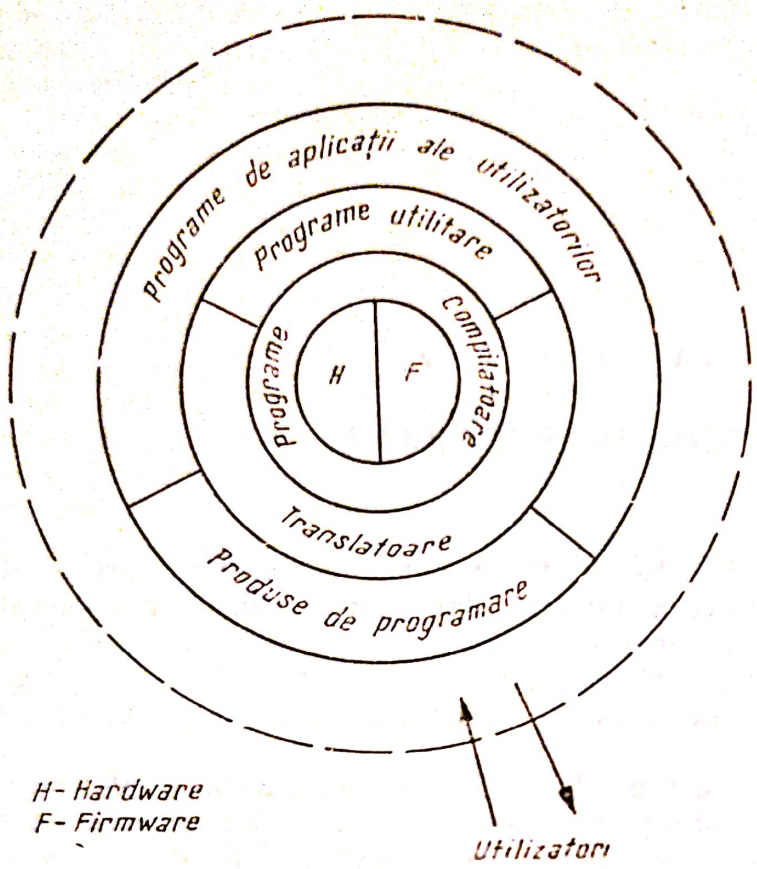


Fig. 12.1.

— sistemele de programe (în sensul definiției date la începutul acestui capitol), care formează așa-zisa parte de „software" ;

— sistemele de microprograme (v. cap. 11), parte numită adesea și „firmware".

În acest mod structura unui sistem de calcul putea fi reprezentată schematic (dar sugetiv !) prin schița din figura 12.1, unde prin produs de programare au fost denumite acele programe care formează unități de sine-stătătoare, cu o utilizare specifică bine definită (parte importantă la calculatoarele de proces).

12.1. SISTEME DE OPERARE

După cum se știe, limbajul de programare reprezintă un element important în procesul de prelucrare a informației în calculatoarele electronice. În acest sens, una din tendințele majore în dezvoltarea tehnicii de calcul este abstractizarea limbajelor de programare prin detașarea lor față de calculatorul electronic pe care sînt implementate. Această idee, cu toate progresele făcute în domeniul tehnicii de calcul, rămîne totuși numai un deziderat al dezvoltării sistemelor de calcul.

O altă direcție importantă în dezvoltarea calculatoarelor o reprezintă aplicarea paralelismului în executarea operațiilor, atît în sensul propriu-zis al echipamentului (efectuarea simultană a mai multor instrucțiuni, accese simultane la memorie, precum și instrucțiuni de intrare/ieșire în paralel) cît și în sensul extern al sistemului prin multiprogramare, multiprelucrare (v. pag. 237) etc.

Una din tendințele moderne în dezvoltarea performanțelor sistemelor de operare o reprezintă microprogramarea prin apariția memoriilor semiconductoare rapide, cu un preț scăzut, fapt ce le face larg accesibile. În acest sens devine posibilă implementarea pe același calculator a mai multoi seturi de instrucțiuni, deci facilități sporite în privința compatibilității.

Scanned with OKEN Scanner

Aceste tendinţe de dezvoltare a tehnicii de calcul au condus la creşterea complexităţii echipamentelor de calcul şi a programelor de sistem prin mărirea numărului de programe-compilatoare şi programe-sistem, precum şi dezvoltarea de sisteme de operare ecomplexe.

Definiţia sistemului de operare

Prin sistem de operare se înţelege un complex de programe care permit exploatarea unui sistem de calcul prin execuţia lanţurilor de programe, asigurînd şi o comunicare eficientă între calculator şi utilizator.

Astfel, un sistem de operare prin funcţiunile sale de coordonare a transferurilor de informaţii între unitatea centrală, unitatea de memorie şi periferice, gestionarea fişierelor de date, compilarea programelor, emiterea de mesaje de eroare în cazul greşelilor sintactice sau logice, simplifică foarte mult munca programatorului.

O altă caracteristică a sistemului de operare este de a coordona resursele sistemului de calcul în vederea satisfacerii la maximum a cererilor utilizatorilor într-un timp de răspuns cît mai scurt.

În cazul aplicaţiilor în timp real (proces), sistemul de operare trebuie să ţină seama de unele restricţii care apar cum ar fi apariţia aleatoare a evenimentelor exterioare şi timpul limitat de răspuns.

Structura sistemului de operare

În cadrul acestui paragraf este reprezentată schematic structura unui sistem de operare, descrierea şi funcţionarea componentelor mai principale ale acestuia, pentru aplicaţiile de proces (timp real). În figura 12.2. este

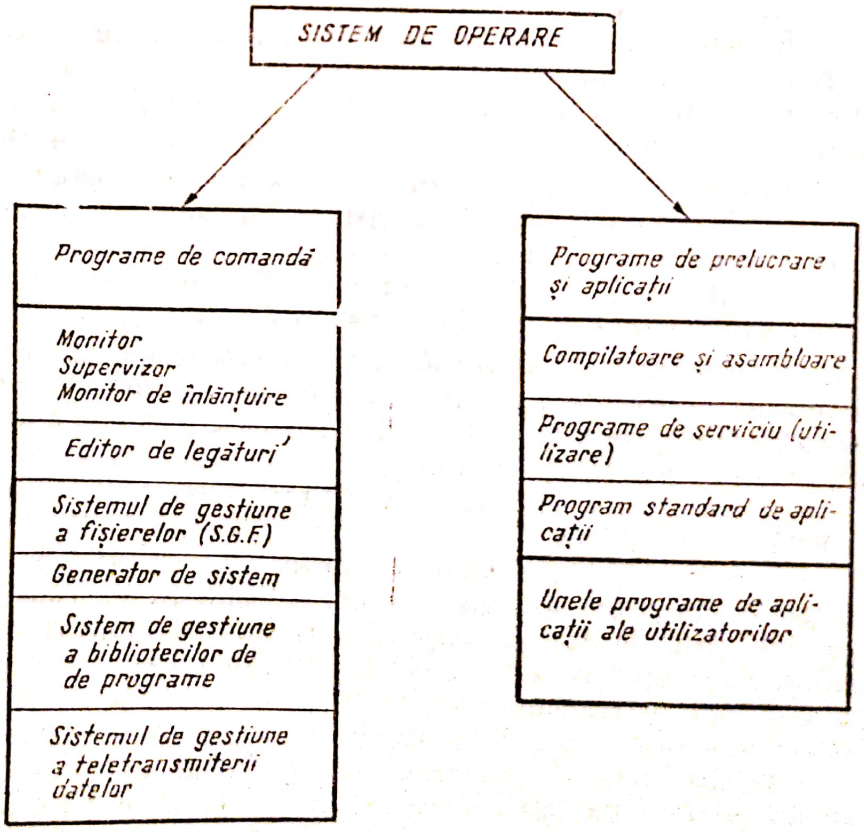


Fig. 12.2.

schematizată o astfel de structură tipică de sistem de operare, la care, în cazuri particulare, o serie de funcţiuni pot lipsi sau din contră pot fi prezente.

De asemenea, trebuie menţionat faptul că datorită volumului mare de programe ce alcătuiesc sistemul de operare, memoria internă a sistemului de calcul fiind limitată, spaţiul acordat pentru memorare permanentă (rezident) este limitat. În acest spaţiu rezident se vor afla programe care stau permanent cum este supervizorul de intrare/ieşire, programe de comandă cu regim privilegiat care pot face apel la instrucţiuni privilegiate (intrare/ieşire) precum şi zone de manevră.

Pentru programele de aplicaţie, este afectată o zonă de memorie în afara părţii rezidente, limita între aceste două zone putînd fi fixă sau variabilă.

Funcţiunile principale ale sistemului de operare

Funcţiunea de administrare a datelor reprezintă una din principalele funcţiuni ale unui sistem de operare privind administrarea datelor în executarea programelor utilizatorilor. Pentru realizarea acestei funcţiuni, de care depinde fiabilitatea faţă de programele utilizatorului, un sistem dispune de resurse fizice (hard) şi de resurse logice (soft).

În cadrul acestei funcţiuni, sistemul de operare presupune asigurarea intrărilor/ieşirilor, a comenzilor mecanice necesare acţionării directe asupra echipamentului periferic, ceea ce reprezintă administrarea fizică a datelor, iar din punctul de vedere al administrării logice a datelor, asigură funcţiuni destinate utilizării resurselor logice prin punerea la dispoziţie a unor rutine de stocare şi regăsire a informaţiei.

Funcţiunea de planificare reprezintă funcţiunea care asigură planificarea lucrărilor în sensul asigurării şi administrării fluxului de lucrări ce sînt prezentate la intrarea în sistem, precum şi planificarea sarcinilor care asigură iniţierea executării de module constituite ca sarcini independente.

Funcţiunea de organizare a bibliotecii de programe este o funcţie asigurată de o componentă a sistemului de operare denumită bibliotecar. Acest program are rolul de a cataloga diferite programe refolosibile de către utilizatori, aflate în oricare din formele sursă, liniar-translatabil (BT) sau imagine memorie translatabilă (IMT).

Funcţiunea de editare de legături este asigurată de către un program denumit editor de legături, care constituie în majoritatea sistemelor de operare un program independent. În cazul unor sisteme de calcul mici, se preferă evitarea programului de editor a legăturilor realizîndu-se compilatoare ce preiau şi funcţia de editare, exemplu — calculatorul românesc FELIX C—32 U/P.

Funcţiunea de gestionare a liniilor de teletransmisie a datelor reprezintă o parte importantă a sistemului de operare în cazul calculatoarelor de proces, prin faptul că implică activităţi suplimentare specifice de recunoaştere şi apelare a terminelelor (traductoarelor). Mesajele şi datele care sosesc pot fi transmise caracter cu caracter sau în blocuri, realizîndu-se o conversie de coduri între calculator şi liniile de telecomunicaţii.

Pentru realizarea acestei funcţii, sistemele de operare au incluse o componentă denumită sistemul de gestiune a teletransmisiilor. În cazul

Scanned with OKEN Scanner

calculatorului de proces de construcţie românească FELIX C—32P, sistemul de operare are inclus, în cadrul programelor monitor, un program monitor timp real.

Afară de aceste funcţiuni, sistemele de operare mai au incluse o serie de funcţiuni puse la dispoziţia programatorilor cum ar fi compilatoarele, ansambloarele, programe utilitare şi aplicaţii standard.

Din considerente economice, un sistem de operare pentru un sistem de calcul nu poate fi acelaşi pentru toate configuraţiile de echipamente ce îl alcătuiesc. Din acest motiv, constructorul creează un sistem standard de referinţă (SSR), după care fiecare sistem de calcul, în funcţie de echipamentele pe care le are în componenţă, îşi creează un sistem de operare adecvat.

Fazele de pregătire ale unui program

La realizarea unui program pentru un sistem ce lucrează în timp real (calculatoare de proces) este necesară o cunoaştere foarte amănunţită a procesului respectiv, a modelului şi a strategiei de conducere, a aparatajului de măsură şi de reglare din componenţa lui. Pe baza acestor elemente, programatorul va realiza un set de programe (scrise într-un limbaj evoluat) denumite programe sursă, pentru care există un compilator în sistemul de operare al calculatorului. În această primă fază se stabilesc părţile majore ale unui program general de conducere al unui proces, cuprinzînd funcţiile de conducere, intrările şi ieşirile de date, dialogul cu operatorul, gestionarea zonelor de memorie la dispoziţia utilizatorului.

O a doua fază o reprezintă compilarea şi corectarea erorilor de sintaxă apărute pe parcursul programului. Aceste programe compilatoare sînt apelate sistemului de operare prin intermediul unor cartele de comandă. În urma acestei faze programul sursă va fi împărţit în module obiect al căror număr este egal cu numărul de componente ce îl alcătuieşte. De menţionat că în această fază programul compilat nu este executabil.

Faza de compilare se va repeta pînă cînd apare o listă fără erori de sintaxă (fig. 12.3) după care se trece la

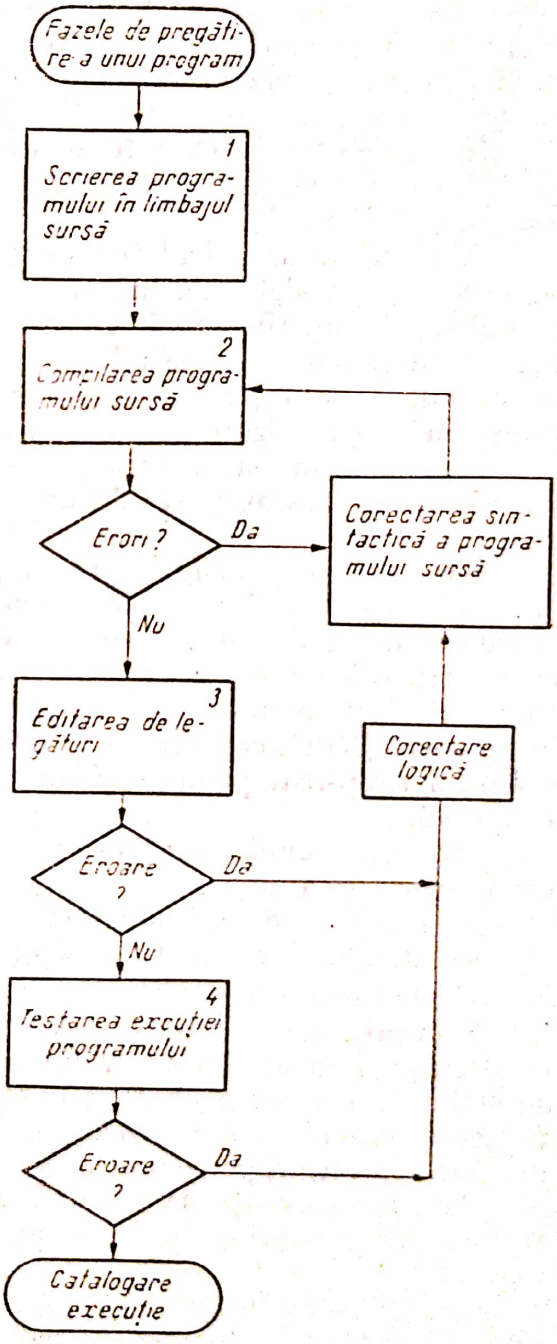


Fig. 12.3.

faza următoare care este obligatorie, editarea de legături. Acest program are rolul de a lega logic modulele obiect elaborate şi de a stabili unele adrese în memorie. În această fază sînt detectate erorile de logică a programului, erori ce trebuie să fie înlăturate pentru a putea fi lansat în execuţie. În cazul unor calculatoare de proces cu capacitate mică de memorare, în momentul apelării programului de compilare, dacă programul nu a avut erori de sintaxă, compilatorul va chema automat programul de editare de legături.

O ultimă fază de pregătire a programului o reprezintă testarea lui, ceea ce înseamnă execuţia programului cu date de încercare. Procedura de alegere a metodologiei de testare depinde de tipul programului, de algoritmul ales şi de modul de exploatare. În acest sens se întocmesc programe de încercări, se elaborează date de încercare şi se stabilesc exemple de control.

În momentul în care toate aceste faze sînt realizate, programul sau pachetele de programe pot fi puse în exploatarea curentă pentru realizarea scopului final.

În cazul unui calculator de proces, un program timp-real se poate desfăşura în paralel cu o lucrare serie, programul timp-real fiind prioritar.

Programele timp-real se încarcă în partiţiile lucrărilor catalogate, pentru ca sistemul să permită executarea instrucţiunilor privilegiate. Din acest motiv, la iniţializarea sistemului este necesar să se declare partiţie programelor catalogate ce va fi ocupată de un program timp-real.

Pentru a putea fi încărcat şi lansat un program timp-real, acesta trebuie să se găsească catalogat în biblioteca utilizator standard (BUS) de pe discul sistem.

Interfaţa monitor/lucrul în timp real se realizează prin funcţiile standard ale monitorului şi prin funcţiile specifice timpului real. Funcţiile standard ale monitorului sînt la dispoziţia programelor timp real la fel ca pentru orice program obişnuit, în plus monitorul prin modulele proprii timpului real realizează funcţiile de activare-dezactivare a nivelelor de întrerupere (NIT), conectarea şi deconectarea întreruperilor externe la un program timp-real, precum şi punerea în stare inactivă a unui program timp-real.

Structura unui program timp-real poate fi compusă din trei tipuri de secvenţe şi anume:

— secvenţe de iniţializare;
— una sau mai multe secvenţe de întrerupere;
— una sau mai multe secvenţe întîrziate.

Secvenţa de iniţializare este obligatorie într-un program timp-real, ea desfăşurîndu-se sub controlul monitorului pentru timp real. În cadrul acestei secvenţe se efectuează o serie de operaţii, cum ar fi afectarea perifericelor, deschiderea fişierelor, conexiunea programelor de întrerupere la nivelele de întrerupere externă.

Această secvenţă se termină cu o macroinstrucţiune care cere monitorului plasarea programului în stare inactivă, în aşteptarea evenimentelor externe.

Secvenţa de întrerupere este lansată de o întrerupere externă şi trebuie prelucrată în timpul cel mai scurt de la apariţia ei. În acest scop, secvenţele de întrerupere sînt lansate direct de către hardware, fără

Scanned with OKEN Scanner

intervenţia monitorului. De aici rezultă că aceste secvenţe de întrerupere nu trebuie să facă apel la funcţiile standard ale monitorului (pentru evitarea perturbărilor unui alt utilizator) sau la el însuşi, prin modificarea stării registrelor.

La încheierea secvenţei de întrerupere, acesta va reda controlul monitorului, indicîndu-i nivelul de întrerupere corespunzător secvenţei, pentru ca acesta să dezactiveze nivelul şi să preia un mod de lucru normal sub controlul monitorului.

Secvenţele întîrziate apar în cazul în care într-un program timp real sînt prelucrări ce nu pot fi efectuate în secvenţele de întrerupere, fie că sînt prea lungi, fie că acestea reclamă funcţii standard ale sistemului, fie o combinaţie între cele două.

Pentru asigurarea interfeţei programator/timp-real, sistemul de operare timp real (monitorul) comportă un anumit număr de funcţii ce permit conectarea secvenţelor de întrerupere la întreruperile externe, deformarea nivelelor de întrerupere, activarea şi dezactivarea nivelelor de întrerupere, punerea programului în stare inactivă precum şi lansarea unei secvenţe întîrziată.

Secvenţele de iniţializare precum şi secvenţele întîrziate trebuie să se termine printr-o macroinstrucţiune (v. macroinstrucţiuni) ce permite să pună programul timp-real fie într-o stare inactivă, fie de a lansa o secvenţă întîrziată cerută în prealabil.

Ex. 12.—1. **Program scris în limbajul de asamblare LAF.** Conţine un program complet scris în limbajul de asamblare LAF pentru calculatorul de proces FELIX C—32. Programul conţine cele trei secvenţe obligatorii pentru lucrul în timp real şi anume: secvenţa de utilizare, secvenţa de întrerupere şi secvenţa întrîziată.

Particularităţile sistemului de operare pentru calculatoarele de proces

Sistemul de operare pentru calculatoarele de proces, avînd la bază lucrul în timp real, în afara funcţiilor generale ale unui sistem de operare, are unele particularităţi datorită funcţionării continue a legăturii proces-calculator. Necesitatea unei astfel de funcţionări non-stop implică existenţa unui program de coordonare denumit program monitor pentru timp real, care trebuie să îndeplinească următoarele funcţiuni:

— controlul calculatorului, mai precis tratarea întreruperilor şi a devierilor;
— dialogul operator-calculator;
— tratarea intrărilor/ieşirilor de date;
— încărcarea programelor şi lansarea lor;
— tratarea funcţiilor de servici (de exemplu, conversia);
— gestiunea zonelor de memorie;
— tratarea firelor de aşteptare;
— multiprogramarea.

Sistemul monitor pentru timp real (MTR) are în componenţa sa o parte principală rezidentă în memoria principală a calculatorului, într-o zonă stabilită pe tot timpul funcţionării calculatorului. În afara acestei

Scanned with OKEN Scanner

```
DATA:11/11/11   H:12332114     PAG.    1

*       ***********************************
*       EX.12.1. PROGRAMAREA COLOR TREI
*       SECVENTE NECESARE UNUI PROGRAM
*       TIMP REAL SI ANUME :
*           -SECVENTA DE INITIALIZARE;
*           -SECVENTA DE INTRERUPERE;
*           -SECVENTA INTIRZIATA.
*       ACESTE SECVENTE SINT PROPRII
*       CALCULATORULUI ROMINESC FELIX
*       C-32 SI SINT PROGRAMATE IN
*       LIMBAJUL DE ASAMBLARE LAF  32.
*       ***********************************

DATA:11/11/11   H:12332776     PAG.    2

                CSECT
SINCR           DATA,4,4    SECVAST
ADRMES          DATA,4,4    X'00020000'
ORG             DATA,4,4    X'FFFFECED'
REF             DATA,4,4    X'FFFFECED'
CELORG          DATA,4,4    0
SALV            RES,4       1
ZECE            DATA,2,2    10
F0              DATA,2,2    X'F0F0'
*
*   SECVENTA DE INITIALIZARE
*   ***********************
START           RES,4       0
                ATTACH      IL:X'90',XZ:ORG,CNT

                ATTACH      IL:X'A8',XZ:ADITORG

                LD4,0       %
                SH4,0       16
                STH2,0      ADITORG+8
                RETURN      ST:TABSINCR
```

```
32          *  BLOC DE SINCRONIZARE
33          *  **********************
34          TABSINCR  RES,4       0
35                    SB          SINCR
36          *  ZONA DE SCHIMB
37          *  **************
38          *
39          ADITORG   RES,8       2
40                    DATA,4,4    ITORG
41                    DATA,4,4    X'00E000ZA'
42          *
43          *  SECVENTA DE TRATARE A INTRERUPERILOR
44          *  ************************************
45          ITORG     RES,4       0
46                    ST4,0       SALV

47                    LD4,0       REF
48                    AD4,0       ORG
49                    ST4,0       ORG
50                    IC4,10      CELORG
51                    LD4,0       SALV
52                    DEACTIV     XZ:ADITORG,SB:SINCR


53          *  SECVENTA INTIRZIATA
54          *  *******************
55          SECVAST   RES,4       0
56                    LD4,0       ORG
57                    DVU2,0      ZECE
58                    SLL4,0      8
59                    SRL2,0      8
60                    SLL4,0      8
61                    MG2,0       F0
62                    STU2,0      ADRMES+2
63                    TYPE        BOKS

64                    WAIT
65                    LD4I,0      50
66                    CP4,0       ORG
67                    BLEZ        TERM
```


Scanned with OKEN Scanner

```
DATA:11/11/11   H:12340162    PAG.

68                  RETURN     ST:TABSINCR

69        TERM      RES,4      0
70                  DETACH     IL:X'90'

71                  DETACH     IL:X'A8'

72                  TERM
73        BCMS      TWB,0      ADRMES+1

74                  END        START
IEC3:  ***  SINT 0000 ERORI  ***  SINT  43 INSTRUCTII  ***

DATA:11/11/11   H:12340868    PAG.
```

părţi, sistemul de operare MTR mai are o serie de procesoare şi o bibliotecă de programe.

Rolul părţii rezidente, care conţine supervizorul şi monitorul propriu-zis, este de a conduce calculatorul pentru a realiza toate funcţiunile arătate mai sus.

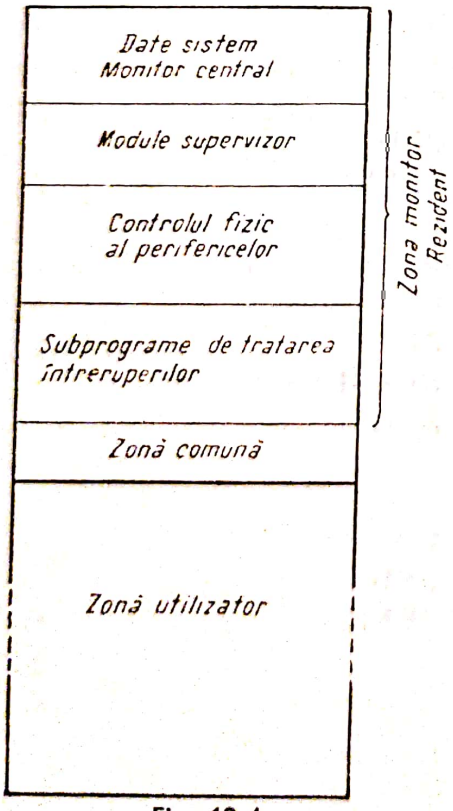


Fig. 12.4.

În general, structura memoriei unui calculator de proces este cea din figura 12.4, şi anume:

— zona monitorului propriu-zis care cuprinde în general adrese de comunicaţie, zona de memorie de salvare în cazul derutărilor, adrese de tabele sistem, adrese de comunicaţie între modulele supervizor, constante sistem, tabele cu intrări/ieşiri de date;

— zona modulelor supervizor;

— zona controlului fizic de transfer al perifericelor;

— zona modulelor de programe caracteristice întreruperilor şi anume: programe de tratare a întreruperilor de la pupitrul de comandă şi programe de tratare a întreruperilor externe;

— zonă comună;

— zona utilizator.

În continuare sînt descrise unele particularităţi ce se impun componentelor unui

sistem de operare pentru timp real, şi anume programelor monitor central (director), supervizor, de gestiune a liniilor de teletransmisie precum şi de diagnosticare.

Programul monitor central (program director). Programul monitor central (MC) sau programul director, cum mai este denumit, are rolul de a supraveghea lucrul în calculator, ţinînd seama de necesităţile acestuia. Astfel, programul MC examinează succesiv starea mesajelor şi a funcţiunilor sistemului, dînd controlul programelor ce se găsesc în sistemul MTR, după nevoi.

Astfel, în figura 12.5 este schematizată funcţionarea unui program monitor, în decursul lucrului în timp real, precum şi nivelele de întreru-

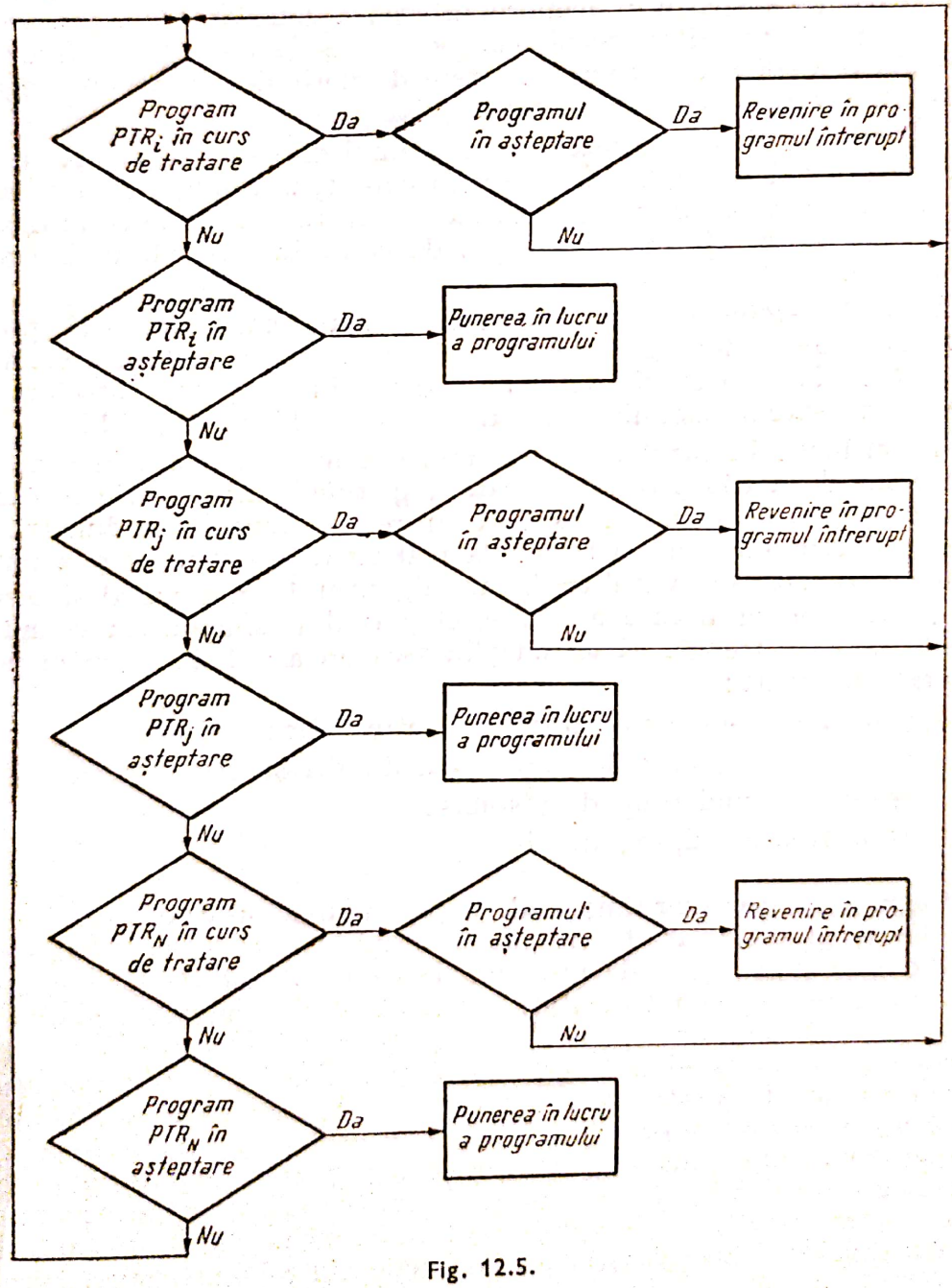


Fig. 12.5.

Scanned with OKEN Scanner

pere. Programele preiau comanda de la monitor şi pot fi întrerupte în cazul unui program cu prioritate mai mare.

Programul în timp real (PTR_i), după cum se vede în figura 12.5, are prioritatea cea mai mare şi deci în timpul tratării lui nici un alt program nu poate fi executat.

Există, de asemenea, programe PTR_j cu o prioritate mai mică, ele putînd fi întrerupte de către programele PTR_i, reluîndu-şi funcţionarea numai după tratarea acestuia, care are prioritate mai mare.

Deoarece programul monitor este apelat de cîteva ori pe secundă, este de dorit ca el să fie proiectat cît mai scurt posibil pentru a cîştiga timp de execuţie şi spaţiu în memoria internă a calculatorului. Din această cauză, în cazul anumitor evenimente sau mesaje mai puţin frecvente, ce necesită o analiză mai complexă, este de dorit de a avea un program monitor de excepţie, separat de bucla principală.

În figura 12.6 este reprezentat programul director, împreună cu programul monitor de excepţie. În cazul în care apar condiţii ce antrenează tratarea excepţională, programul monitor va întrerupe execuţia programului aplicativ ce se află în lucru şi va da controlul procedurii de tratare excepţională.

În cazul sistemelor de calcul în timp real, mai puţin perfecţionate din punctul de vedere al sistemului de întrerupere, gestiunea intrărilor şi ieşirilor de date poate fi încorporată direct în programul director, care are ca primă sarcină examinarea stării canalelor libere (fig. 12.7).

În concluzie, în cazul realizării unui sistem ce lucrează în timp real este necesar de a căuta o simplificare a gestiunii lucrului. Un mecanism de gestiune foarte complex riscă să antreneze numeroase dificultăţi la realizarea sistemului şi în particular la realizarea programelor şi a testărilor. De menţionat este faptul că fiecare algoritm de gestiune al sistemului este necesar pentru fiecare aplicaţie, el putînd fi simplu sau complicat. Sarcinile pe care trebuie să le îndeplinească un astfel de algoritm pot fi enumerate pe scurt:

— realizarea regulilor de prioritate între diferitele faze;
— realizarea regulilor de prezenţă a diferitelor faze;
— realizarea unui timp de răspuns;
— realizarea multiprogramării.

Programul supervizor timp real. Supervizoarele, utilizate pentru calculatoarele de proces, sînt diferite de la o aplicaţie la alta. De asemenea, complexitatea unui astfel de program de comandă este funcţie şi de sistemul pe care este implementat, astfel dacă sistemul este modest, iar numărul de cereri de acces este scăzut, programul supervizor va fi destul de simplu. Din contră — în cazul unui sistem complex avînd implementate un număr mare de programe de aplicaţie, şi de cereri de acces, precum şi un timp de răspuns foarte scăzut, supervizorul va fi destul de complex. Rolul supervizorului constă în coordonarea şi girarea intrărilor/ieşirilor de date, organizarea firelor de aşteptare atît pentru mesaje cît şi pentru date, tratarea întreruperilor şi a devierilor, alocarea memoriei, controlul şi diagnosticarea în cazul unei funcţionări anormale. Un rol important al supervi-

Scanned with OKEN Scanner

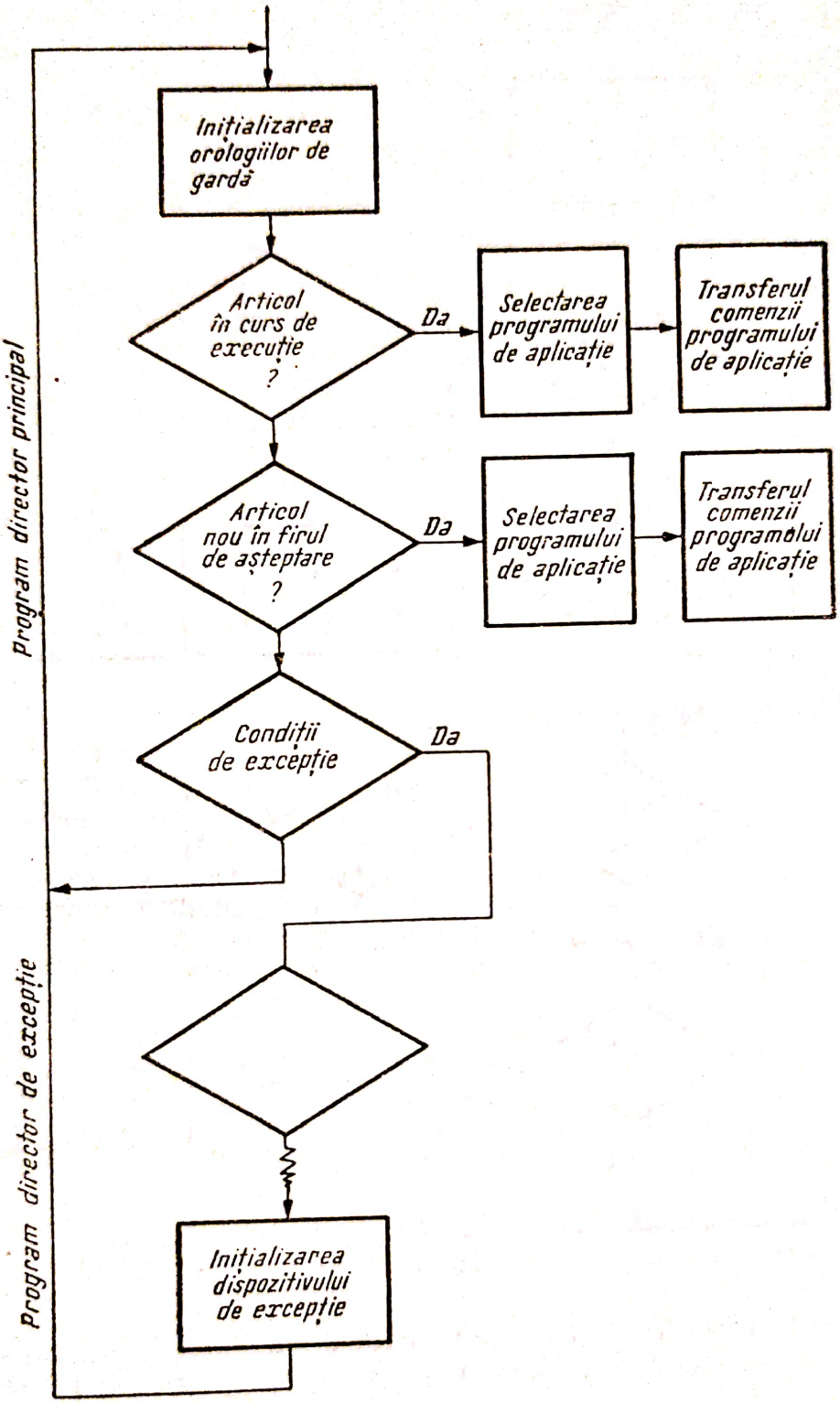


Fig. 12.6.

Scanned with OKEN Scanner

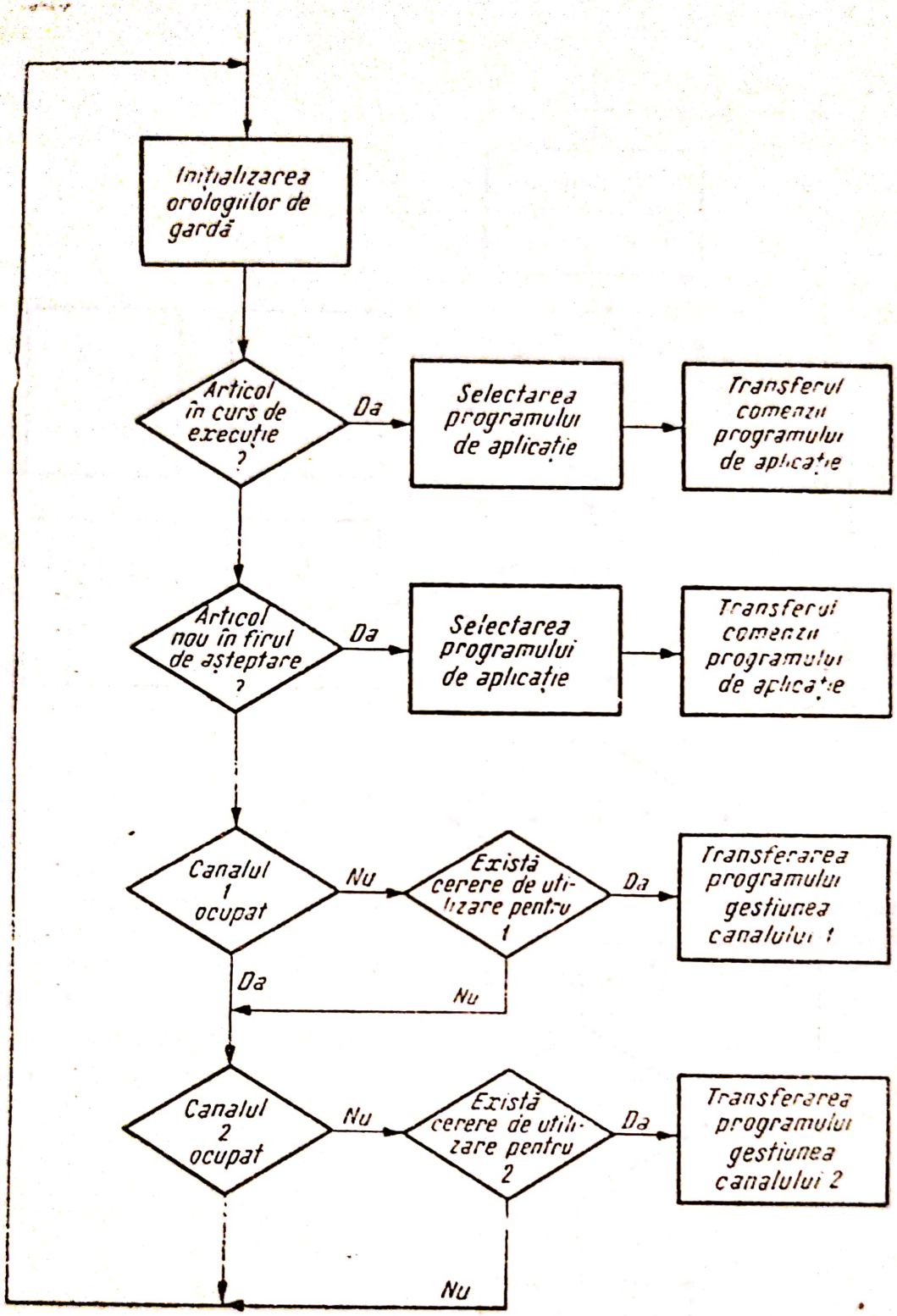


Fig. 12.7.

zorului ce lucrează în timp real este acela de atribuire a priorităţilor programelor de aplicaţie şi de a efectua funcţiile de punere în lucru a acestora.

Descrierea funcţionărilor supervizorului în timp real. În continuare, pentru a înţelege mai bine funcţiile supervizorului să considerăm un sistem de calcul ce lucrează în timp real în legătură directă cu un proces. În

Scanned with OKEN Scanner

momentul sosirii unui semnal pe linia de comunicaţie, supervizorul va trebui să încarce în memorie programul sau subprogramul ce tratează acest tip de semnal. Secvenţa operaţiilor pe care le-ar executa în acest caz supervizorul ar putea fi următoarele:

— întreruperea activităţii normale de calcul şi testarea dacă un mesaj sau un semnal este în aşteptare pentru a fi tratat;

— în cazul în care un mesaj sau semnal se află în aşteptare este transmis un ordin către memoria centrală împreună cu codul de selecţionare a programului de tratare;

— programul selecţionat este încărcat în memoria calculatorului în locuri disponibile fără a distruge ceea ce există înaintea întreruperii;

— tratarea mesajului sau a parametrului şi elaborarea de răspunsuri sau comenzi dacă sînt necesare;

— preluarea comenzii de către programul întrerupt, în cazul că nu există un alt semnal sau mesaj prioritar în tratare.

În cazul unui volum mare de mesaje sau parametri, al unui timp de răspuns foarte scăzut, al unei fiabilităţi şi al unui număr mare de decizii, programul supervizor devine destul de complicat.

În rezumat, a fost dată mai jos o listă — de funcţiuni necesare programului supervizor, în cazul unui sistem complex şi anume:

— comanda intrărilor/ieşirilor în sensul planificării operaţiilor unităţilor de intrare/ieşire şi a canalelor de transmisie, controlul şi buna funcţionare a lor;

— comanda liniilor de comunicaţie, apelarea şi controlul terminalelor, controlul erorilor;

— verificarea şi transmiterea de mesaje;

— gestiunea unităţilor de imprimare, vizualizare etc.

— comunicarea cu operatorul;

— gestiunea funcţiilor calculatorului;

— gestiunea firelor de aşteptare;

— alocarea memoriei diferitelor funcţiuni în cazul unei comenzi;

— comunicarea de date între calculatoare în cazul unei reţele de calculatoare;

— gestiunea întreruperilor (în acest sens se analizează cauza provocării întreruperii, transferînd controlul programului prioritar cel mai apropiat, conservă registrele de lucru şi datele programului întrerupt pentru a putea fi reluat);

— salvează datele, registrele, în cazul trecerii normale de la un program la altul;

— controlul acţiunilor declanşate de orologiul intern al calculatorului;

— asigură securitatea fişierelor în cazul utilizării programelor de test sau în cazul programelor eronate;

— gestiunea diagonsticurilor în timpul funcţionării sistemului pentru asigurarea unei securităţi depline.

Macroinstrucţiuni. Pentru a transfera comanda de la un program de aplicaţie la programul monitor (director), programatorul utilizează un set de comenzi denumite macroinstrucţiuni, care după compilare sînt transformate în mai multe instrucţiuni cod maşină. O parte din macroinstruc-

Scanned with OKEN Scanner

țiunile folosite, împreună cu anumite subprograme de bibliotecă necesare compilării, fac parte din ansamblul de instrucțiuni ale programului supervizor.

În cazul unei programări clasice a unui sistem de gestiune a intrărilor/ieșirilor de date, programatorul va utiliza la alegere, prin intermediul unui set de macroinstrucțiuni, subprograme existente sau scrise de el. Din contră în cazul programării în timp real a unui sistem de gestiune a intrărilor/ieșirilor de date, nu va mai fi o alegere de subprograme, deoarece toate intrările/ieșirile de date sînt girate de către programul supervizor.

Există un timp de macroinstrucțiuni cu un rol important în execuția unui program de aplicație în timp real, propriu programului monitor (director). În continuare vor fi descrise diferite tipuri de macroinstrucțiuni caracteristice monitorului central (deci monitorului supervizorului) împreună cu funcțiile pe care le îndeplinesc.

Macroinstrucțiunea ASTEAPTA (WAIT, ATTENTE) are rolul de a pune un program de aplicație în așteptare datorită unor cauze interne ale sistemului de operare. De exemplu, în cazul în care un program de aplicație face apel la un fișier și nu poate continua lucrul cu el, programul nu va bloca calculatorul, datorită acțiunii acestei macroinstrucțiuni. Adresa programului de aplicație precum și condițiile de întrerupere sînt înregistrate într-o zonă de memorie, de exemplu în blocul de referire a mesajelor (BRM), după care predă comanda programului monitor (director). La fiecare cerere de intrare/ieșire, suprogramul supervizor ce se ocupă cu aceasta, va mări cu o unitate valoarea contorului situat de exemplu în BRM, după care valoarea lui este diminuată cu 1 după fiecare operație. În momentul trecerii controrului prin zero, programul director poate din nou să dea comanda programului de aplicație.

Macroinstrucțiunea IEȘIRE (SORTIE, OUTPUT) este ultima fază a unui program de aplicație, semnificînd că tratarea lui este terminată.

După această macroinstrucțiune, programul supervizor va repune la zero toate informațiile cuprinse în zonele de lucru (informații cu privire la programul de aplicație ce a fost executat, de exemplu în BRM) și toate mesajele vor fi șterse în vederea unui nou program. După această macroinstrucțiune putem controla valoarea contorului pentru intrări/ieșiri de date, și în cazul că valoarea lui este diferită de zero, această informație va fi înregistrată în zona de mesaje pentru următoarea execuție, după care controlul va fi transmis programului monitor (director) pentru o nouă tratare.

Macroinstrucțiunea AFECTARE MEMORIE este utilizată în momentul nevoii de spații în memorie pentru tratarea diferitelor mesaje. Dacă memoria centrală a calculatorului este împărțită în blocuri dinamice comune, în cazul apariției unui mesaj, un astfel de bloc va fi pus la dispoziția lui, caz în care numărul blocurilor disponibile va fi diminuat cu o unitate.

Pentru a nu apare riscul ca sistemul să rămînă fără nici un bloc de memorie liber, există un subprogram supervizor care are rolul de a superviza aceste alocări, el putînd termina tratarea unui mesaj în curs de execuție sau de a transfera pe disc anumite module ale programului de aplicație existent în memorie. De menționat că în momentul derulării acestui subprogram, alocarea sau recuperarea blocurilor de memorie, precum și apariția de întreruperi sînt interzise.

Scanned with OKEN Scanner

Macroinstrucţiunea DESAFECTAREA MEMORIEI reprezintă inversul macroinstrucţiunii descrise mai sus.

Macroinstrucţiunea INTRARE/IEŞIRE este executată de un supraprogram, supervizor standard, fazele de execuţie fiind descrise în cadrul programului supervizor.

Această macroinstrucţiune realizează toate posibilităţile de intrare/ieşire de date:

— scrierea/citirea discului magnetic;
— scrierea de pe disc combinată cu o aşteptare;
— scrierea/citirea pe o bandă magnetică, rebobinarea ei;
— imprimarea la maşina de scris, imprimantă sau consolă de vizualizare;
— imprimarea mesajelor, sau transmiterea lor către multiplexor.

Programul supervizor ce execută aceste macroinstrucţiuni face apel mereu la programul de canal pentru a vedea dacă acestea sînt disponibile.

Macroinstrucţiunea de creare a unui BLOC DE REFERIRE A MESAJELOR are rolul de a crea o zonă în memorie, un tabel cuprinzînd toate detaliile necesare tratării unui mesaj în timpul execuţiei unui program de aplicaţie.

În continuare se mai pot cita cîteva tipuri de macroinstrucţiuni care participă la execuţia unui program de aplicaţie cum ar fi macroinstrucţiunea de legătură între diferite programe, macroinstrucţiunea de întreţinere şi punere la zi etc.

Avînd în vedere posibilităţile macroinstrucţiunilor descrise, programatorul unei aplicaţii va fi ajutat în munca sa de programare. Ca exemplu, se poate da execuţia intrărilor/ieşirilor de date ce este realizată de către programele supervizor prin intermediul macroinstrucţiunilor de intrare/ieşire. Totuşi în execuţie apar unele restricţii, inevitabile, deoarece mai multe programe împart acelaşi sistem de calcul. Aceste restricţii variază de la un sistem la altul, ele devenind precise pentru acela care cunoaşte ansamblul supervizor. În continuare putem cita cîteva din aceste restricţii şi anume:

— un program de aplicaţie în execuţie nu poate face apel la zone de memorie care nu i-au fost afectate;
— nici un program de aplicaţie nu poate da ordine de efectuare de intrare/ieşire de date sau întreruperi, acestea fiind realizate de macroinstrucţiunile arătate;
— conservarea conţinuturilor registrelor de lucru utilizate în cadrul programelor de aplicaţie, este temporară;
— limitarea, pentru fiecare program de aplicaţie a timpului de utilizare a unităţii centrale, prin intermediul orologiilor interne ale calculatorului, pentru a nu depăşi mai mult de 500 milisecunde între două macroinstrucţiuni succesive AŞTEAPTĂ sau IEŞIRE;
— în cazul transmiterii unui mesaj, acesta poate fi decupat în mai multe părţi datorită faptului că zona tampon are lungime mai mică, el fiind redat în mai multe faze (în figura 12.8 este reprezentat un exemplu de tratare a unui mesaj în timpul execuţiei unui program de aplicaţie).

Cazul cel mai tipic este acela de a avea mesajul decupat în două înregistrări. Pentru a executa această fază, programul are nevoie de spaţiu

Scanned with OKEN Scanner

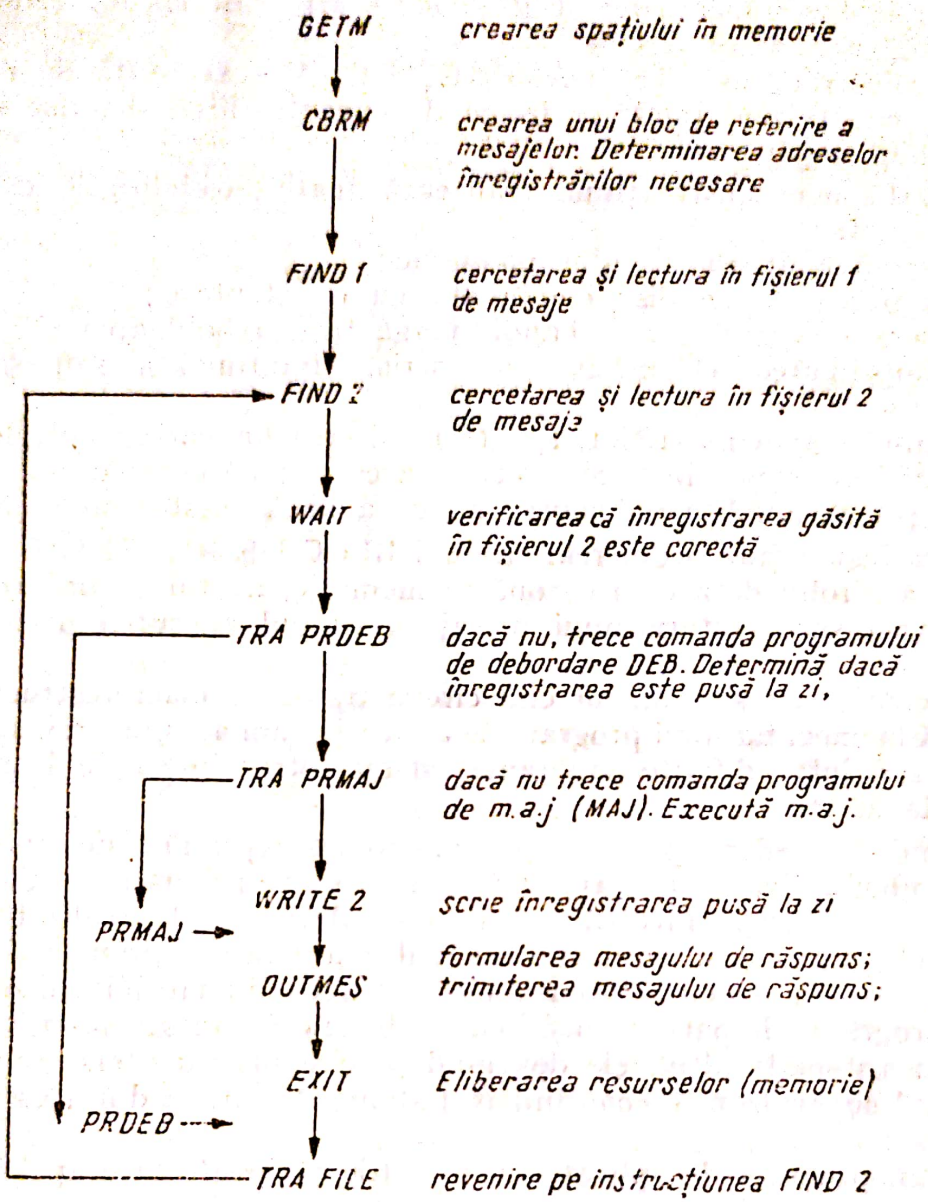


Fig. 12.8.

în memorie și de alcătuire a zonei de referire a mesajelor (BRM), aceasta făcîndu-se cu ajutorul a două macroinstrucțiuni, descrise mai sus. În continuare se vor calcula adresele înregistrărilor, în acest fel putîndu-se executa lectura. Această căutare a înregistrărilor în vederea lecturii, după cum se vede în figura 12.8, se face cu ajutorul unei macroinstrucțiuni denumite FIND, care este urmată de o macroinstrucțiune WAIT, ce va permite sistemului de a executa altceva în timpul căutării.

Programul va prelua comanda în momentul citirii înregistrărilor, va face controlul și validarea lor și în cazul în care acestea nu sînt bune, va căuta o zonă în memorie pentru a formula mesajul. După expedierea mesajului, programul va utiliza o macroinstrucțiune EXIT, care va elibera zonele de memorie utilizate, pentru a putea fi tratate și alte mesaje, după care dă controlul programului supervizor.

Scanned with OKEN Scanner

În cazul calculatorului de proces FELIX C—32 P, de fabricație românească, se pot exemplifica o serie de macroinstrucțiuni timp real ce permit efectuarea de programe timp real și anume:

— macroinstrucțiunea SB (bloc de sincronizare) are rolul de a determina într-o secvență de întrerupere dacă a fost sau nu cerută lansarea unei secvențe întîrziate;

— macroinstrucțiunea ATTACH permite completarea unui cuvînt de întrerupere cu adresa zonei de schimb destinată lansării unei secvențe de întrerupere; de asemenea, validează și armează nivelul de întrerupere corespunzător;

— macroinstrucțiunea DETACH interzice lansarea unei secvențe de întrerupere, dezarmează și invalidează nivelul de întrerupere corespunzător;

— macroinstrucțiunea ACTIV permite generarea unei întreruperi prin program;

— macroinstrucțiunea DEACTIV are rolul de a dezactiva nivelul de întrerupere;

— macroinstrucțiunea RETURN permite fie să pună programul timp-real în stare inactiv (în așteptarea unui eveniment), fie de a lansa o secvență întîrziată cerută în prealabil printr-o macroinstrucțiune DEACTIV.

Programul de gestiune a liniilor. Pentru o unitate de intrare/ieșire de tip bandă magnetică sau disc magnetic, programul principal din memorie primește și prepară un mesaj, după care un subprogram scrie sau citește mesajul și controlează dacă schimbul s-a executat corect. În cazul unui program de gestiune al liniilor de transmisie, acesta este mult mai complicat, deoarece o informație este desfăcută în biți și transmisă bit cu bit la viteza de transmitere a liniei. La sfîrșitul transmisiei, biții se vor ansambla în caractere și caracterele în mesaj. Este necesar de a se detecta pe cît posibil și a se corija erorile asupra cuvintelor, apărute în cazul transmiterii. În afara mesajelor, calculatorul mai transmite în timp util și semnale de comandă pentru terminalele la distanță. Toate aceste funcțiuni de transmitere a informațiilor și gestionare a liniilor sînt asigurate în diverse maniere și anume fie prin programe de gestiune a liniilor, fie prin circuite cablate sau o combinație între programe de gestiune a liniilor și circuite cablate.

Funcțiile de execuție a acestor programe ce asigură o legătură între liniile de comunicație și programele principale ale ordinatorului pot fi următoarele:

— declanșarea și girarea datelor la recepție, provenite din transmiterea pe liniile de telecomunicații; în cazul mai multor linii de telecomunicație, acestea pot emite simultan, iar terminalele pot fi explorate sau apelate selectiv;

— asamblarea biților în caractere și caracterele în mesaje;

— convertirea codului de transmisie în codul calculatorului în cazul codurilor diferite;

— detectarea erorilor de transmisie, ce pot apare în transmiterea caracterelor și a mesajelor;

— editarea mesajelor, în cazul tapării greșite de către operator;

Scanned with OKEN Scanner

— recunoaşterea caracterelor de sfîrşit de mesaj ; executarea operaţiilor corespunzătoare şi prepararea pentru o altă transmitere ; în cazul detectării unui mesaj eronat este necesar să se realizeze retransmiterea ;

— transmiterea mesajelor programului principal, codificate unul după altul. În această situaţie este necesar a se avea în vedere girarea firului de aşteptare ce se formează ;

— prepararea mesajelor pentru transmitere către terminale, prin decodificarea din cod calculator în codul de transmitere pe linie, precum şi adăugarea de caractere de control ;

— girarea procesului de transmitere şi repetarea mesajelor sau a caracterelor în cazul detectării unei erori de transmitere ;

— semnalarea sfîrşitului de transmisie către terminal.

În cazul unui sistem de calcul ce lucrează în timp real, care nu are decît o singură legătură cu un terminal, controlul transmisiei este relativ simplu, deoarece liniei i se va rezerva o singură zonă tampon în memoria centrală, zonă ce va fi umplută cu caractere transmise. În cazul în care linia permite o transmitere biunivocă, atunci se vor rezerva două zone tampon.

Problema se complică în cazul mai multor terminale, deci a mai multor linii de transmitere, deoarece este posibil ca toate să emită sau să primească în acelaşi timp.

Pentru început, programul de gestiune a liniilor va atribui la fiecare linie cîte o zonă de memorie, unde se vor stoca, caracter cu caracter datele primite. În momentul umplerii unui bloc, i se va asocia prin înlănţuire un al doilea bloc, ş.a.m.d. pînă cînd primim tot mesajul (figura 12.9), ultimele caractere din bloc conţinînd adresa blocului următor.

Esenţialul în gestionarea liniilor de transmisie este de a indica fiecărui terminal momentul cînd poate să emită sau să recepţioneze mesaje, comenzi etc. Deoarece terminalele, în cazul aplicaţiilor industriale, nu transmit într-o manieră aleatorie, acestea pot fi interogate la un interval de timp stabilit, funcţie de necesităţile procesului condus, aceasta fiind realizată prin intermediul unui dispozitiv de întrerupere pentru înaltă prioritate, declanşat de orologiul de teletransmisie.

Programe de diagnostic. Rolul programelor de diagnostic este acela de a controla buna funcţionare a calculatorului, de a încerca să repare eventualele pene de funcţionare şi de a semnala în caz de nereuşită.

Erorile care apar în funcţionare pot fi clasificate ca erori permanente, acele erori ce apar de fiecare dată cînd se verifică circuitul corespunzător, şi erori intermitente, erorile ce apar aleatoriu. Acest tip de eroare este destul de dificil, deoarece programele de detectare pot trece peste ele.

În cazul producerii unei erori, întreruperea provocată trece controlul de la programul în curs la un program de diagnostic care va urmări una din următoarele variante :

— în cazul unei erori trecătoare, programul va încerca să execute din nou instrucţiunea ce a declanşat eroarea şi să continue tratarea ;

— repararea erorii nu este posibilă, dar programul poate fi memorat pentru o eventuală refacere şi reluare ;

— în cazul distrugerii de date sau a unor părţi de program, se va trece automat la reîncărcarea programului ;

Scanned with OKEN Scanner

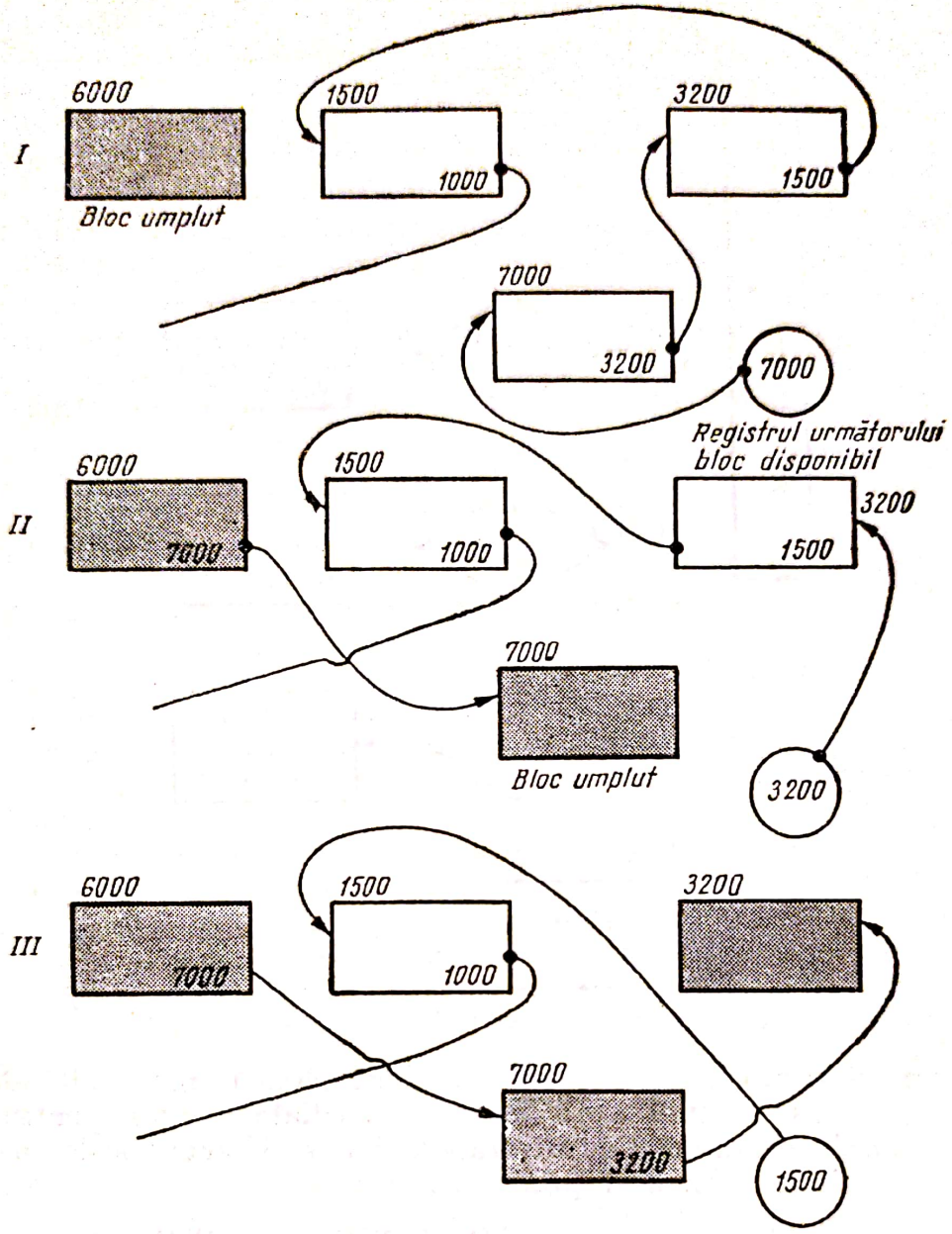


Fig. 12.9.

— în cazul unei pene permanente detectată la sistemul de calcul, se va face salvarea lucrului, apoi oprirea calculatorului.

În figura 12.10 sînt arătate posibilităţile şi nivelele de reluare a execuţiei unui program de aplicaţie. Se vede că în cazul unei reparări imediate a unei erori detectate programul poate fi reluat din punctul de oprire. În cazul în care eroarea nu poate fi detectată imediat, reluarea programului se va face dintr-un punct de reluare superior cel mai apropiat de punctul de întrerupere. Cînd eroarea apărută se datoreşte sistemului de calcul şi acesta necesită o reparare, reluarea se va face de la începutul programului.

Scanned with OKEN Scanner

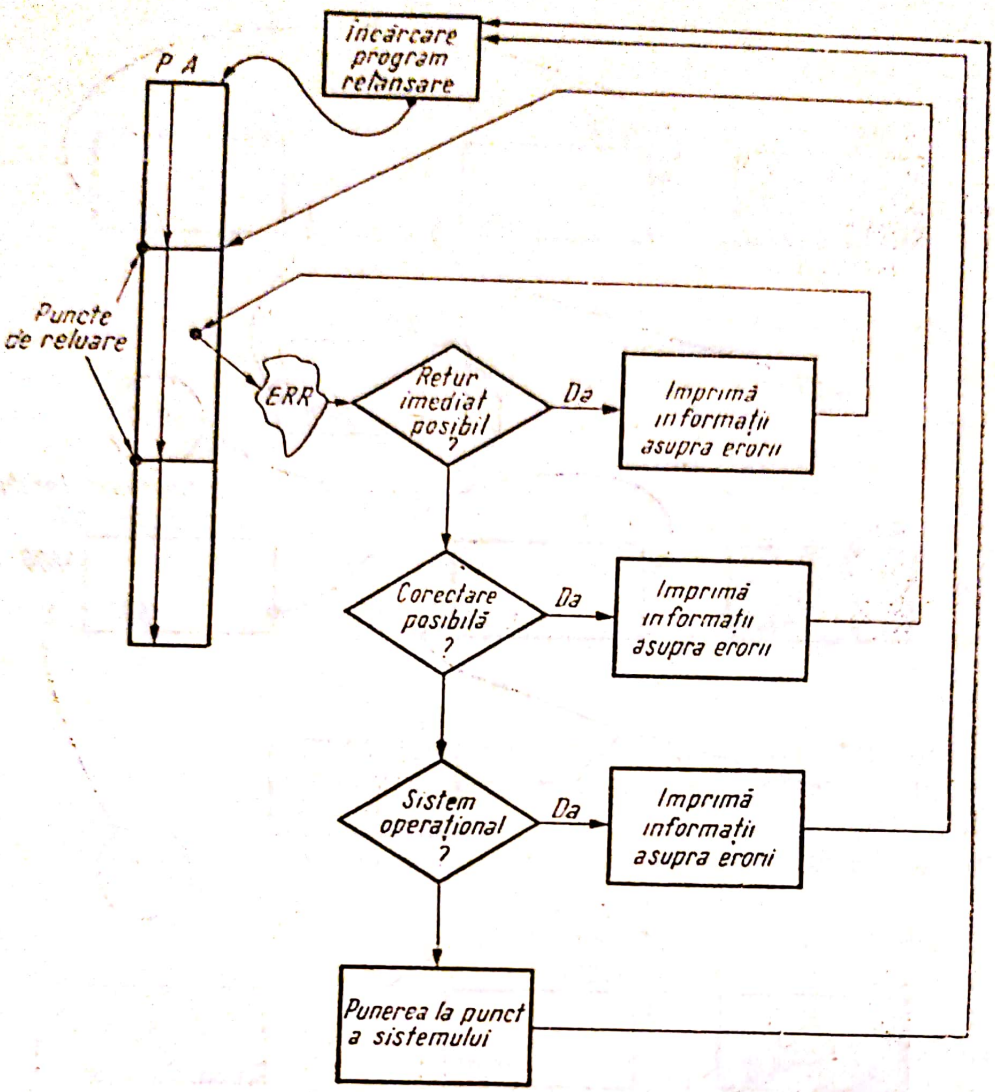


Fig. 12.10.

Pentru a avea o corectare automată sînt necesare mai multe elemente şi anume, un sistem eficace de detectare imediată a erorii pentru lucrul timp real, un sistem automat de întrerupere timp real, un mijloc de conservare a programului în momentul producerii întreruperii.

Exemple de sisteme de operare (FELIX C—32P, IBM—1800)

Ex. 12—2. **Calculatorul FELIX C—32P.** În cadrul conducerii unui proces tehnologic, calculatorul de proces FELIX C—32P (v. ex. 11.1) poate realiza următoarele funcţiuni :

— supravegherea şi controlul centralizat, prin asigurarea urmăririi centralizate a parametrilor din proces, într-o ordine dictată de explorare (în cazuri de erori se asigură imprimarea unui mesaj de alarmă) ;

— regim ghid operator în care calculatorul are rol de consultant privind comenzile şi manevrele necesare pentru desfăşurarea corectă a procesului ;

— comanda mărimilor de referinţă ale regulatoarelor ;

— studierea procesului tehnologic în vederea stabilirii modelului matematic ;

Scanned with OKEN Scanner

— optimizarea procesului, prin determinarea valorilor optime ale parametrilor din proces conform unui criteriu ales.

Sistemul de exploatare și programare al calculatorului FELIX C—32P, după cum este arătat în fig. 12.11 este alcătuit din două sisteme de exploatare și anume:

— sistemul de exploatare de bază SFC—32
— sistemul de exploatare pentru proces PROSIS—32.

În cazul sistemului de exploatare de bază, acesta este compus din: monitor, sistemul de gestiune al fișierelor, SGF, programe traducătoare — — standard.

Sistemul de exploatare pentru proces, după cum este arătat în fig. 12.11, cuprinde următoarele programe de bază:

— Sistemul de programare DELIS compus din:
— procesor DELIS;
— procesor pentru limbajul mnemonic DELIS—LMD;
— procesor de pregătire standard a protocolării PRGS.
— modulul de gestiune a orologiilor MGO;
— modulul de comandă a secvențelor întîrziate COMSIN;
— sistemul de programare pentru aplicații de proces SPAP compus din următoarele programe standard:
— 3 programe de explorare și prelucrare PG, E, PS;
— program de protocolare standard PPR;
— program de deservire a consolei operatorului de proces—PSERCOP;
— program de deservire a cererilor de proces prin mașina de scris — — PSERMAS;
— sistemul de programe de testare a echipamentului — STES.

Sistemul de programare DELIS este un sistem de programare de tipul fill-in-the-blanks specializat pentru conducerea proceselor industriale. La baza acestui sistem de programare se găsesc trei procesoare și anume DELIS, LMD, PRGS, care lucrează în regim off-line și care generează un sistem de fișiere EDFIS pe baza informațiilor primite de la utilizator.

Acest fișier creat va conține toate informațiile necesare exploatării sistemului de programe de aplicații proces SPAP.

Procesorul DELIS realizează prelucrarea informațiilor furnizate de utilizator cu ajutorul limbajului DELIS.

Pe baza informațiilor referitoare la variabilele de proces se poate dezvolta modelul de reglare dorit și se prelucrează un set de cartele de comandă, care permit atît procesorului DELIS cît și sistemului SPAP desfășurarea unei activități standard indiferent de proces.

Procesul LMD are la bază limbajul mnemonic DELIS-LMD, conceput pentru a permite utilizatorului scrierea unor programe scurte, necesare unor acțiuni specifice care nu se pot standardiza și care sînt executate de sistemul SPAP.

Procesorul PRGS are rolul de a defini configurația de protocolare necesară derulării programului de protocolare standard PPR.

Modulul de gestiune al orologiilor — MGO pune la dispoziția sistemului și utilizatorului două categorii de orologii și anume: orologii hardware și orologii software. La dispoziția utilizatorului se găsesc 8 orologii programate comandate de orologiul hardware.

Scanned with OKEN Scanner

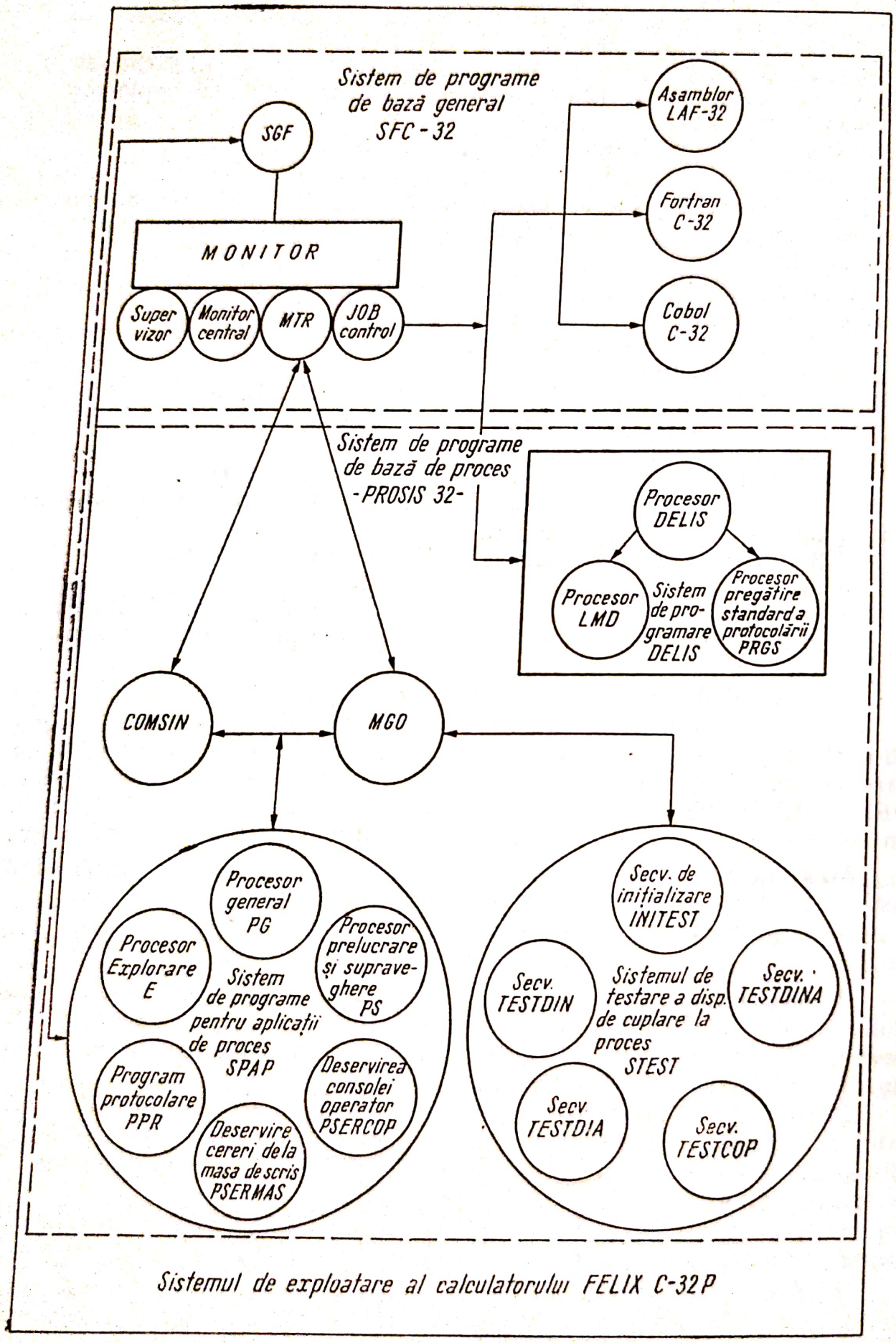


Fig. 12.11.

Scanned with OKEN Scanner

Cu ajutorul acestor orologii se obţin timpi de ordinul milisecundelor, iar cu ajutorul orologiilor programate se obţin timpi de ordinul secundelor, minutelor, orelor.

Modulul de comandă a secvenţelor întîrziate realizează comanda secvenţelor întîrziate ale SPAP, apelînd la funcţiile monitorului timp real MTR.

Sistemul de programare pentru aplicaţii SPAP este un sistem de secvenţe de întrerupere şi secvenţe întîrziate ce permit supravegherea procesului.

Procesul general PG, ce este inclus în sistemul SPAP, este un program de iniţializare ce permite conectarea nivelelor de întrerupere, deschiderea fişierelor de lucru, iniţializarea unor tabele şi zone de lucru, iniţializarea valorii curente a variabilelor de proces şi a valorilor prescrise.

Procesorul de explorări are ca rol citirea şi memorarea valorii mărimilor explorate la intervalele prestabilite de utilizator, validarea informaţiilor citite, filtrarea numerică precum şi comanda efectivă a valorilor de referinţă.

Ultimul procesor din componenţa SPAP este procesorul de prelucrare şi supraveghere PS, care realizează prelucrarea variabilelor cu prioritate înaltă şi prelucrarea periodică a variabilelor la intervale de timp stabilite de utilizator, prin identificarea variabilelor, conversie, calcule speciale asupra valorii curente, verificări asupra limitelor, calculul valorii prescrise, precum şi alte acţiuni propuse de utilizator.

După cum am amintit mai înainte, în componenţa acestui sistem SPAP mai intră şi un număr de programe, şi anume:

— programul de protocolare PPR care are ca rol de a imprima periodic istoricul procesului, realizînd un protocol pentru variabilele de proces;

— programul de serviciu pentru consola operatorului, PSERCOP, este un program specializat ce realizează dialogul în timp real dintre operator şi calculator. Acest program are ca sarcină de a supraveghea calculatorul de proces, de a afişa valorile, de a schimba starea unor instrumente de măsură şi regulatoare, de a modifica unele informaţii din fişiere, de a apela unele programe şi de a testa elementele de afişare;

— programul — PSERMAS — de servire a cererilor de proces este un program ce realizează dialogul timp real între operator şi calculator prin intermediul maşinii de scris (MAE).

Acest program asigură tipărirea de valori ale unei variabile de proces, modificarea unei valori, schimbarea stării unor instrumente, apelarea unor programe, modificarea unor constante scrise în limbajul LMD.

Sistemul de programe de textare a echipamentului STES este un set de programe timp real care permite testarea cuplorului de proces. El verifică funcţionarea fiecărui dispozitiv de intrare analogică DIA, dispozitiv de intrare numerică DIN sau o combinaţie analogico-numerică DINA, pentru toate intrările/ieşirile numerice şi analogice indicate. Acest sistem de programe asigură selectarea dispozitivului pe intrare/ieşire, lansarea în execuţie a secvenţei de test, analiza informaţiei primite, imprimarea la MAE a rezultatului operaţiei de testare precum şi supravegherea dialogului operator/sistem. În componenţa sistemului intră o serie de secvenţe cum ar fi secvenţa de iniţializare, testare pentru DIA, DIN, DINA, precum şi secvenţa de testare a funcţionării corecte a consolei operatorului.

Scanned with OKEN Scanner

Ex. 12—3. **Calculatorul IBM—1800.** Pentru realizarea funcţiilor sale (v. Ex. 11.3) calculatorul beneficiază de un sistem de programe denumit PROSPRO—1800, care este conceput şi dezvoltat pentru a standardiza şi simplifica controlul programat al proceselor industriale cu ajutorul calculatorului de proces.

Pentru generarea sistemului de programe PROSPRO sînt necesare următoarele faze:

— generarea sistemului pe disc;
— memorarea bibliotecilor de programe PROSPRO;
— generarea programelor de proces;
— simularea programelor on-line;
— memorarea cartelelor de date PROSPRO.

Structura sistemului PROSPRO—1800 este formată dintr-o serie de programe specializate şi anume:

— *Procesing Scheduler Program* (PSP) (programul de proces) care realizează comanda generală a calculatorului. El selectează programele majore ce vor fi executate pe baza priorităţilor. După executarea fiecărui program, comanda este preluată mereu de programul PSP, pentru o nouă prelucrare.

— *Ajustment Procesor* (AP) (programul de ajustări).

— *General Procesor* (GP) (procesorul general) care are ca scop identificarea acţiunilor generale şi de a colecta informaţiile necesare paşilor specifici acţiunii. O acţiune generală este o secvenţă specifică din paşii acţiunii standard care este interpretată şi executată *în linie* în timpul procesului general de reglare.

— *Acţiunea generală* este un program unic, definit într-un limbaj specific generalizat care permite executarea unui număr de calcule şi decizii, executate fără a recurge la un program FORTRAN, limbajul fiind alcătuit din cîteva coduri ce definesc cerinţele speciale ale procesului. Paşii acţiunii sînt interpretaţi şi executaţi secvenţial, începînd cu primul pas şi continuînd pînă la sfîrşitul acţiunii sau pînă cînd programul este dirijat către o altă acţiune.

— *Variable Procesor* (VP) (procesorul de variabile) este un program care utilizează datele din file de informaţii asupra variabilei.

Moduri de tratare a lucrărilor utilizate de calculatoarele de proces

Problemele practice care se pun spre rezolvare cu ajutorul calculatoarelor de proces sînt de o mare diversitate. De la funcţia de reglare numerică directă, care solicită o frecvenţă ridicată a comunicaţiilor proces-calculator-proces, pînă la cele mai diverse probleme care nu au legătură cu procesul condus, denumite curent probleme nonproces, aria problemelor tratate cu calculatoarele de proces este foarte largă. Pentru a face faţă acestor solicitări, au fost elaborate diferite modalităţi de tratare a lucrărilor, care, în funcţie de specificul aplicaţiilor efective, tind spre optimizarea utilizării resurselor fizice şi logice ale calculatoarelor de proces.

Tratarea secvenţială. Deşi acest mod de lucru este tot mai puţin utilizat, şi cu atît mai puţin în cazul calculatoarelor de proces, el prezintă importanţă pentru înţelegerea celorlalte moduri de tratare a lucrărilor. Modul secvenţial este cel mai simplu şi, istoric, primul mod de tratare a lucrărilor.

Scanned with OKEN Scanner

În cadrul modului de tratare secvențial, lucrările sînt executate una după alta, în ordinea sosirii acestora în unitatea centrală. Sistemul de operare asigură înlănțuirea automată a lucrărilor, eliminînd necesitatea încărcării programelor în calculator prin acțiuni ale operatorului. Tratarea secvențială a lucrărilor presupune formarea unor șiruri de lucrări și are drept consecință obținerea unui timp de răspuns lung la solicitările utilizatoruluilor.

Multiprogramarea. Aceasta definește un mod de funcționare al sistemului, conform căruia un calculator execută simultan mai multe lucrări [1,39]. Aplicarea acestui mod de funcționare a pornit de la observația că un program de aplicații nu consumă, în general, toate resursele pe care le are la dispoziție. Astfel, în memoria internă pot exista simultan mai multe programe care trebuie să fie executate. Atunci cînd un program particular ajunge la un punct de oprire, un alt program, deja existent în memoria internă, poate fi executat de către unitatea centrală. În acest mod se economisește timpul care ar fi necesar pentru încărcarea de la un anumit periferic a acestui program.

În figura 12.12 este prezentată schematic tehnica multiprogramării. Programele P_1, P_2, și P_3 se găsesc în memoria internă. La momentul t_0, unitatea centrală este ocupată cu prelucrarea primului program UC(P_1),

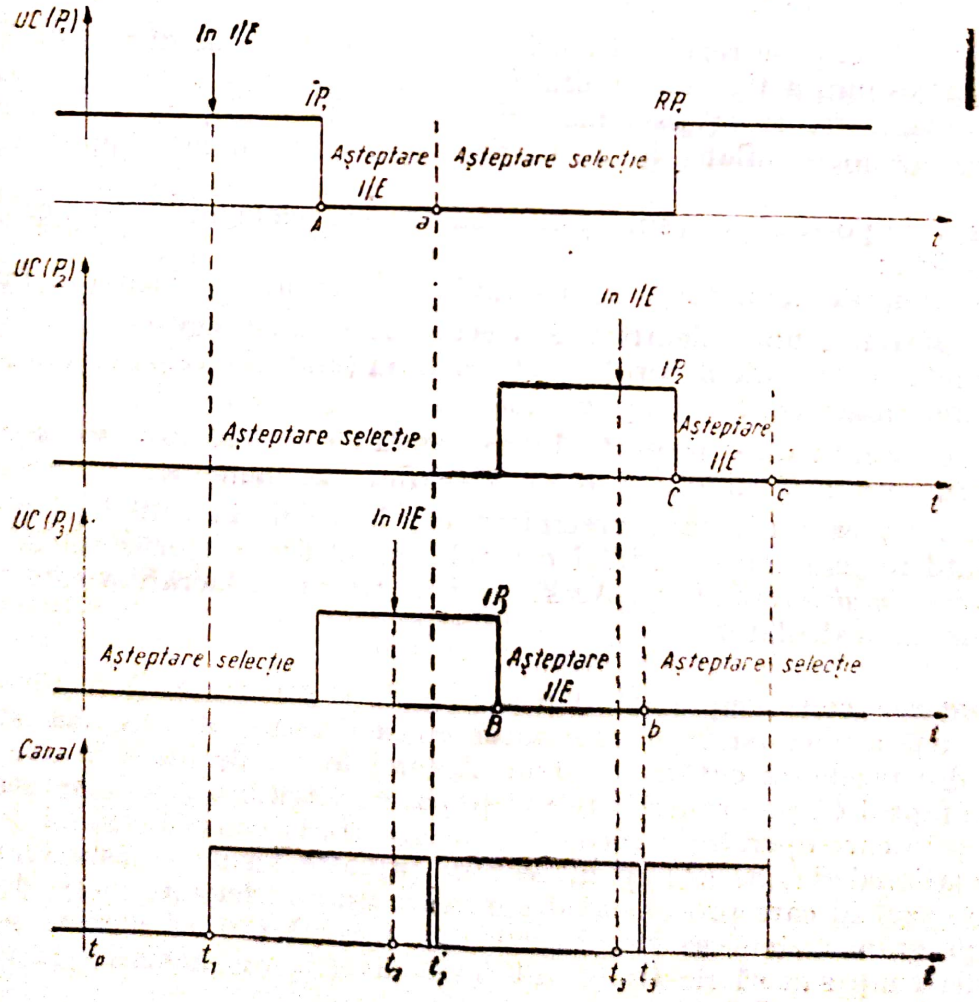


Fig. 12.12.

care la timpul t_1 inițiază o operație I/E. Canalul fiind liber execută această operație I/E. imediat. La momentul A, unitatea centrală întrerupe prelucrarea programului P_1, deoarece operația I/E solicitată nu s-a terminat; acesta intră în așteptarea terminării operației I/E.

Presupunînd prioritățile în ordinea P_1, P_3, P_2, în momentul întreruperii programului P_1, algoritmul de selecție inițiază prelucrarea programului P_3 de către unitatea centrală, UC(P_3). Programul P_3 inițiază la momentul t_2 o operație I/E; deoarece canalul este ocupat, aceasta va începe la timpul t_2' cînd se termină operația I/E solicitată de programul anterior P_1.

Execuția programului P_3 este întreruptă la momentul B deoarece operația I/E solicitată nu s-a terminat. În continuare este executat programul P_2.

La momentul de timp $a = t_2'$, cînd s-a terminat operația I/E pentru programul P_1, acesta intră în așteparea unei noi selecții deoarece unitatea centrală este ocupată cu prelucrarea programului activ (în cazul de față, programul P_3).

Din figura 12.12 se observă că unitatea centrală este activă permanent, prelucrînd unul din cele trei programe. Există o singură situație, nefigurată în schemă, cînd unitatea centrală poate deveni inactivă și anume atunci cînd toate programele aflate în memorie așteaptă terminarea unei operații I/E.

Din cele expuse rezultă că prin multiprogramare se urmărește utilizarea la maximum a timpului unității centrale. Valoarea acestui maxim este cu atît mai mare cu cît este mai mică frecvența apariției situației în care toate programele aflate în memorie așteaptă terminarea unor operații I/E.

Este important de subliniat următoarele elemente specifice multiprogramării:

— prezența simultană a mai multor programe în memoria internă;

— prezența unui algoritm de trecere de la un program la altul astfel încît randamentul calculatorului să fie maxim (unitățile cele mai costisitoare să rămînă inactive un timp cît mai scurt);

— existența unor protecții între programe care să asigure executarea unui program fără alterarea celorlalte aflate în memorie.

În fine, este necesară precizarea că, în cazul în care lucrările sînt executate în mai multe unități centrale, se utilizează termenul de *multiprelucrare (multiprocessing)*. Acest mod de tratare a lucrărilor este specific rețelelor de calculatoare.

Tratarea conversațională. Un sistem de calcul tratează lucrările conversațional atunci cînd permite intervenția operatorului în una sau mai multe din fazele executării acesteia. Acestei forme de lucru îi este caracteristic faptul că programul este în așteptare pentru o perioadă substanțială de timp în care operatorul introduce informația la consolă. Modul de lucru conversațional devine scump în ceea ce privește resursele sistemului, mai ales în cazul în care programul despre care este vorba este mare, deoarece acest program trebuie să stea în memoria internă fără să lucreze perioade lungi de timp sau să fie transferat între nivelele de memorie *(swapping)* în mod frecvent [39].

Scanned with OKEN Scanner

Principial există un *mod conversațional complet*, în care utilizatorul comunică cu programul și invers, programul comunică cu utilizatorul, adică, rezultatele parțiale din program pot apare la consolă și programul poate cere și aștepta o informație care să fie introdusă prin consolă.

În afara acestuia, există și un *mod de lucru semiconversațional*, în care programul poate comunica cu utilizatorul, dar acesta nu poate comunica cu programul în timpul rulării, cu excepția posibilității de a-l elimina din calculator.

În cazul tratării semiconversaționale este de dorit ca lucrarea să fie rulată în *partiție serie*, adică odată încărcată să rămînă în memorie pînă la terminare. Prin aceasta se elimină operațiile de transfer al programului între memorii. Pentru programele lungi, acest mod de lucru nu este însă acceptabil.

Dacă sistemul este multiprogramat sau are mai multe unități centrale se elimină operațiile de transferare a lucrărilor între memoria internă și memoria externă, ceea ce conduce la creșterea eficienței sistemului.

Tratarea conversațională a lucrărilor reprezintă un mod de lucru important pentru calculatoarele de proces. Supravegherea procesului cu ajutorul calculatorului este o aplicație frecventă a acestuia. În cazul acestor aplicații, operatorul solicită în mod curent din partea sistemului valorile anumitor mărimi cum sînt: debite, temperaturi, compoziții etc. sau chiar informații sintetice care să oglindească modul de lucru al instalației (cantități din anumite produse, bilanțuri parțiale sau totale etc.).

Prelucrarea cu acces multiplu. Acest mod de prelucrare este cunoscut curent și sub denumirea *regim de lucru cu divizarea timpului (time-sharing)*.

Prin regim de lucru cu acces multiplu se înțelege utilizarea simultană a unui sistem de calcul de la mai multe terminale, calculatorul executînd lucrări independente, practic simultan, prin interpătrunderea executării acestora în unitatea centrală (fig. 12.13).

Prin intermediul terminalelor conectate se pot solicita probleme diferite, din partea fiecărui utilizator, aceștia putînd apela la toate resursele sistemului. În mod ciclic, sau pe baza altor reguli, fiecărui utilizator i se alocă o anumită cuantă de timp în decursul căreia acesta dispune de toate resursele sistemului de calcul, necesare aplicației sale. După epuizarea timpului alocat, utilizatorul pierde controlul asupra sistemului în favoarea altui utilizator și așa mai departe pînă la servirea tuturor. După terminarea ciclului, primul utilizator primește din nou controlul asupra sistemului, problema sa fiind reluată din locul în care a fost întreruptă.

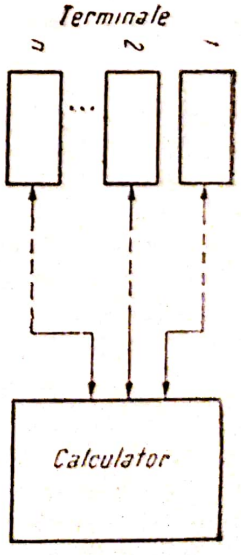


Fig. 12.13.

Există numeroși algoritmi de divizare a timpului între diferiți utilizatori. Unii alocă ciclic o cuantă de timp de mărime constantă tuturor utilizatorilor, alții utilizează scheme cu priorități (de exemplu, se dă prioritate utilizatorilor cu lucrări conversaționale asupra celor cu lucrări lungi). Există și algoritmul care alocă cuante de durate variabile, în funcție de anumite caracteristici ale lucrărilor.

Scanned with OKEN Scanner

Fiecare utilizator intră în interacţiune cu sistemul prin intermediul terminalului la care este conectat. El reprezintă o sursă a unei succesiuni de lucrări ale căror rezultate îi vin înapoi. După executarea unei lucrări, sistemul revine la starea de comandă şi tipăreşte cuvîntul GATA (READY) sau altul echivalent. Din acest moment de timp, utilizatorul poate transmite sistemului o altă lucrare. Din punctul de vedere al sistemului de calcul utilizatorul poate fi privit ca un proces care din timp în timp este oprit şi apoi reactivat. Fiecare din aceste reactivări corespunde executării unei sarcini specifice.

În orice moment de timp, în sistemul de calcul există sub administrarea supervizorului o varietate de procese, unele dintre ele aparţinînd programelor obiect, iar altele sistemului însuşi. Toate aceste procese sînt dispuse într-un număr de *şiruri*, unele conţinînd procese libere pentru execuţie, altele — procese în aşteptare dintr-un motiv sau altul (de exemplu, cele care aşteaptă terminarea unei operaţii de intrare/ieşire).

Alcătuirea şi administrarea şirurilor de lucrări care sînt gata pentru execuţie şi a celor care sînt oprite revin unei componente a supervizorului. Aceasta are la bază un algoritm care aranjează lucrările în diverse şiruri, le transferă dintr-un şir în altul, le ia din şirul lucrărilor libere pentru execuţie şi le activează.

Supervizorul conţine un *algoritm de planificare* care este apelat ori de cîte ori are loc o întrerupere sau derulare, sau un proces în curs de rulare intră în supervizor. Acesta decide dacă se va schimba procesul şi în caz afirmativ, stabileşte care dintre procese să fie activat sau reactivat. Cînd nu există nici un proces liber pentru execuţie, unitatea centrală intră într-un ciclu de *„mers în gol"*.

Algoritmul de planificare are, de asemenea, sarcina de a accepta lucrările care au fost prezentate sistemului, dar nu au fost încă recunoscute de către supervizor. Unele dintre acestea vor fi lucrări off-line sau lucrări executabile în partiţie serie. Acestea sînt introduse sub formă de şiruri în interiorul sistemului, fiind citite şi plasate pe disc de către supervizor, prelucrarea lor activă nefiind totuşi începută. La acestea se adaugă lucrările serie iniţiate din partiţiile paralel şi introduse în şiruri în mod similar. În fine, mai există şi utilizatorii on-line care doresc să se introducă în sistem şi care sînt deserviţi tot de algoritmul de planificare.

Datorită stării diferite a lucrărilor care aşteaptă să fie tratate, este de dorit ca ele să fie introduse în şiruri separate, în conformitate cu originea şi priorităţile lor. Algoritmul de planificare trebuie, pe de o parte, să pună în aplicare strategia privitoare la prioritatea care se ataşează lucrărilor de diverse categorii şi la proporţia de timp al unităţii centrale care li se alocă, iar pe de altă parte, să selecteze astfel lucrările încît să păstreze toate componentele sistemului într-o productivitate cît mai mare posibilă.

Modul de lucru cu acces multiplu a impus noi concepte privitoare la organizarea şi alocarea memoriei. Dintre acestea, *organizarea memoriei pe pagini* şi conceptul de *memorie virtuală* asociat sînt de o mare aplicabilitate.

Conform noii organizări a memoriei interne, aceasta este divizată în blocuri de anumită lungime fixă numite *pagini* (lungimea acestora variază între 512 şi 4 096 bytes).

Fiecare lucrare ce trebuie executată necesită un număr variabil de pagini, care sînt afectate acesteia pe măsură ce execuţia avansează. Iniţial,

Scanned with OKEN Scanner

programul se află memorat pe o memorie externă, de unde este încărcat pagină cu pagină în memoria principală, în funcţie de necesităţi. De aici rezultă avantajele organizării pe pagini ale memoriei principale:

— deoarece încărcarea programelor în memoria principală se face pe pagini, în funcţie de necesităţi, în aceasta vor putea fi prezente pagini din programele multor utilizatori, element esenţial pentru accesul multiplu;

— nefiind necesară încărcarea întregului program în memoria principală, utilizatorul nu este limitat de mărimea fizică a memoriei principale.

Atunci cînd la execuţia unui program se apelează la o instrucţiune care nu se găseşte pe paginile încărcate în memoria principală, *rutina de alocare a memoriei* încarcă pagina lipsă. Pe durata încărcării, unitatea centrală nu rămîne inactivă, execuţia fiind trecută la un alt program prezent în memoria principală.

Programul utilizatorului nemaifiind limitat de mărimea fizică a memoriei interne, rezultă că acesta poate fi considerabil mai mare. Utilizatorul dispune de o memorie virtuală, conform căreia spaţiul ocupat de program este mult mai mare decît spaţiul fizic al memoriei interne.

Prelucrarea în timp real. După cum s-a văzut în paragraful 11.1, calculatoarele de proces sînt conectate prin intermediul terminalelor la procesul supravegheat sau condus. De la acestea, la momente de timp bine precizate sau aleator, în funcţie de evenimentele ce au loc în cadrul procesului, sistemul de calcul primeşte informaţii care trebuie prelucrate. Rezultatele prelucrării trebuie comunicate procesului sau operatorilor într-un anumit timp, denumit *timp de răspuns*.

Stabilirea unei comunicaţii directe în ambele sensuri între proces şi calculator şi prezentarea răspunsului într-un anumit interval de timp constituie trăsăturile de bază ale modului de lucru în timp real.

Durata timpului de răspuns variază în funcţie de specificul procesului şi de caracteristicile sistemului de calcul, acoperind intervalul de la fracţiuni de secundă pînă la cîteva ore.

Din punctul de vedere al prezentării rezultatului, există două cazuri specifice:

— rezultatul prelucrării trebuie dat la momente de timp bine precizate (exemple: perioadele de eşantionare în cazul reglării numerice, orele la care trebuie prezentate protocoalele în cadrul proceselor chimice etc.);

— rezultatul prelucrării trebuie prezentat în interiorul unui anumit interval de timp (exemple: semnalizarea depăşirii limitelor anumitor mărimi din proces, prelucrarea cromatogramelor în vederea determinării compoziţiei etc.).

Pentru a putea răspunde unor asemenea cerinţe, sistemul de operare trebuie să aibă componente care să permită întreruperea execuţiei lucrării în curs şi prelucrarea imediată a operaţiilor solicitate, alocînd imediat resursele fizice necesare (unitate centrală, memorie internă şi periferice). Componenta centrală a sistemului de operare, specifică lucrului în timp real, este *monitorul pentru timp real* care are următoarele funcţiuni principale:

— cercetează periodic cererile curente din sistem;

— pe baza priorităţilor asociate fiecărei cereri decide care este lucrarea ce trebuie executată;

Scanned with OKEN Scanner

— cercetează starea curentă a prelucrărilor, nivelul la care acestea au ajuns şi transferă controlul programului cerut de acestea, pe baza regulilor prestabilite ;

— atunci cînd se termină execuţia unui program, acesta trece controlul coordonatului care decide ce program trebuie executat în continuare.

Algoritmul de planificare conţinut de coordonator este astfel alcătuit încît ţine seama de priorităţile unor lucrări în raport cu altele, timpii de răspuns necesari, disponibilitatea resurselor fizice etc. (v. subcapitolul 12.1).

Din punctul de vedere al apariţiei în timp a lucrărilor ce trebuie executate de un calculator de proces există următoarele posibilităţi :

— lucrări al căror început este stabilit de către operatorul uman ;
— lucrări care încep la momente absolute sau relative bine precizate ;
— lucrări al căror început depinde de evoluţia procesului.

Multiplele aspecte ridicate de lucrările pe care trebuie să le execute un calculator de proces au condus la prezenţa practică a tuturor modurilor de tratare a informaţiei :

— multiprogramarea ;
— tratarea conversaţională ;
— prelucrarea cu acces multiplu ;
— prelucrarea în timp real.

Utilizarea multiprogramării este absolut necesară deoarece aceasta permite executarea simultană de către calculator a mai multor lucrări, aspect ce solicită prezenţa simultană a mai multor programe în memoria internă a sistemului. Lucrările tratate în regimul de multiprogramare pot fi atît din proces cît şi din afara procesului.

Tratarea conversaţională reprezintă o latură specifică calculatoarelor de proces. În mod curent operatorul solicită calculatorului valorile anumitor mărimi tehnologice, intervine pentru a modifica limitele minime şi maxime ale unor variabile, sau valorile parametrilor de acordare în cazul reglării numerice directe etc. Pentru a răspunde acestor solicitări, sistemul de operare trebuie să întrerupă execuţia lucrării în curs şi să reacţioneze în mod corespunzător la cererea operatorului.

Accesul multiplu este absolut necesar în cazul calculatoarelor de proces. Terminalele de la un calculator cu acces multiplu universal sînt înlocuite, în cazul calculatorului de proces, cu terminale specifice informaţiilor ce trebuie culese din proces (traductoare de debit, presiune, temperatură, analizoare etc.). În mod curent, informaţiile generate de traductoare sînt recepţionate de calculator şi prelucrate la anumite perioade. Faţă de această situaţie există şi cazuri de prioritate maximă din partea procesului, care trebuie prelucrate imediat de către unitatea centrală. În asemenea situaţii, sistemul trebuie să pună la dispoziţie toate resursele fizice şi logice necesare.

12.2. PROGRAME DE APLICAŢIE

Prin *programe de aplicaţie* înţelegem acele programe prin a căror execuţie se rezolvă problemele pentru care a fost implementat calculatorul de proces. Corespunzător acestor probleme, mărimea unui program de aplicaţie poate să fie de la cîteva instrucţiuni pînă la solicitarea unui spaţiu de

Scanned with OKEN Scanner

memorare de cîteva ori mai mare decît capacitatea memoriei interne disponibile. În a doua situaţie programul trebuie segmentat şi, segment cu segment, trebuie chemat de pe un suport de memorie externă în memoria internă spre a fi executat.

Pentru a fi executate, programele de aplicaţie apelează la programele ce alcătuiesc sistemul de operare, în special pentru operaţiile de coordonare şi de intrare/ieşire.

Programele de aplicaţie trebuie alcătuite, practic, pentru fiecare calculator de proces implementat în parte, dacă nu complet, cel puţin supuse anumitor modificări. Chiar şi în cazul unei aceleiaşi aplicaţii tehnologice asemănătoare cu una precedentă, dacă se modifică configuraţia calculatorului de proces, atunci programele de aplicaţie trebuie supuse unei prelucrări noi. Programele standard, de exemplu cele pentru funcţiile aritmetice de bază cu operare în virgulă mobilă, precizie dublă, funcţii transcendente etc., rămîn nemodificate. Aceste programe sînt alcătuite într-o formă optimală pentru fiecare calculator de proces şi stau la dispoziţia utilizatorului în aşa-numita bibliotecă de programe.

Tipuri de programe de aplicaţie

Tipul unui program de aplicaţie este precizat prin anumite caracteristici specifice. Acestea se referă la modalităţile de interacţiune cu programele ce compun sistemul de operare şi la subordonarea fiecărui program în parte. În acest sens există *programe principale*, *programe parţiale* şi *subprograme*.

Programe principale. Tipul de program principal este caracterizat prin aceea că un asemenea program are un număr, o prioritate şi un nume. În general, prioritatea scade cu creşterea numărului, adică, un program principal cu numărul 12 este situat ca prioritate în urma celor cu numerele 0 ... 11. Unui program principal i se atribuie de către supervizor un contor propriu de comenzi. De aici rezultă că un program principal este administrat direct de către sistemul de operare, prin aceasta înţelegîndu-se toate operaţiile de coordonare, adică, începerea şi sfîrşitul execuţiei, prelucrarea în funcţie de priorităţi, întreruperile, transferările între memoria internă şi cea externă etc.

Numele are, pe lîngă importanţa formală pentru compilator, scopul de a permite chemarea nominală respectiv de a pune la dispoziţie programe. Acest lucru este foarte util, în primul rînd, atunci cînd se foloseşte stocarea programelor de aplicaţii pe suporturi de memorie externă.

Programele principale sînt alcătuite pentru rezolvarea efectivă şi completă a unei probleme. Un program principal ar putea fi şi un *program independent*, care, la un anumit moment din execuţia sa necesită informaţii de la un alt program principal, sau generează informaţii pentru un alt program principal. De fapt, în cazul implementării unui calculator de proces nu există programe principale independente. Există numai *programe principale dependente*, care depind unul de altul şi îşi transferă date într-o măsură mai mare sau mai mică. Coordonarea acestora este realizată de sistemul de operare.

Scanned with OKEN Scanner

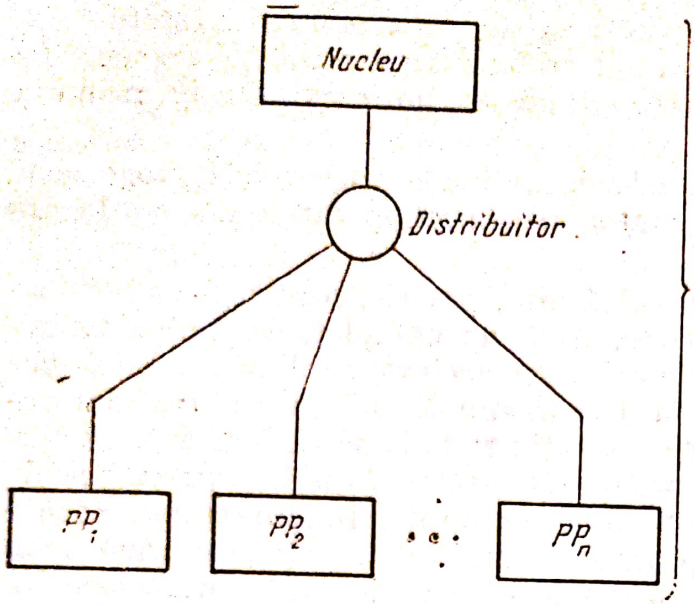


Fig. 12.14.

Programe parțiale. Din considerente privind organizarea programului, împărțirea memoriei interne disponibile, ușurința înțelegerii acestuia etc., programele principale de dimensiuni apreciabile sînt divizate în așa-numitele programe parțiale. Cu ajutorul programelor parțiale se rezolvă anumite părți distincte din cadrul întregii probleme care face obiectul programului principal. Caracteristic pentru programul parțial este modul în care se realizează coordonarea acestuia și anume aceasta nu se realizează de către sistemul de operare ci de către programul principal din care face parte. Prioritatea programelor parțiale corespunde celei a programului principal.

Programele parțiale PP_1, PP_2, ..., PP_n sînt conectate cu *nucleul programului* principal prin intermediul unui *distribuitor* (fig. 12.14). Nucleul preia în acest caz funcția de coordonare a programelor parțiale.

Există două modalități de organizare a unui program de aplicație. Într-unul din cazuri, întregul program principal, adică nucleul și programele parțiale, este stocat în memoria externă. La startul programului principal, nucleul singur sau nucleul și un program parțial (întotdeauna același) se încarcă în memoria internă (fig. 12.15, *a*). Celelalte programe parțiale sînt încărcate de către nucleul programului, după necesități. De obicei, programul parțial nou chemat în memoria internă ocupă spațiul de memorie ocupat de programul parțial anterior.

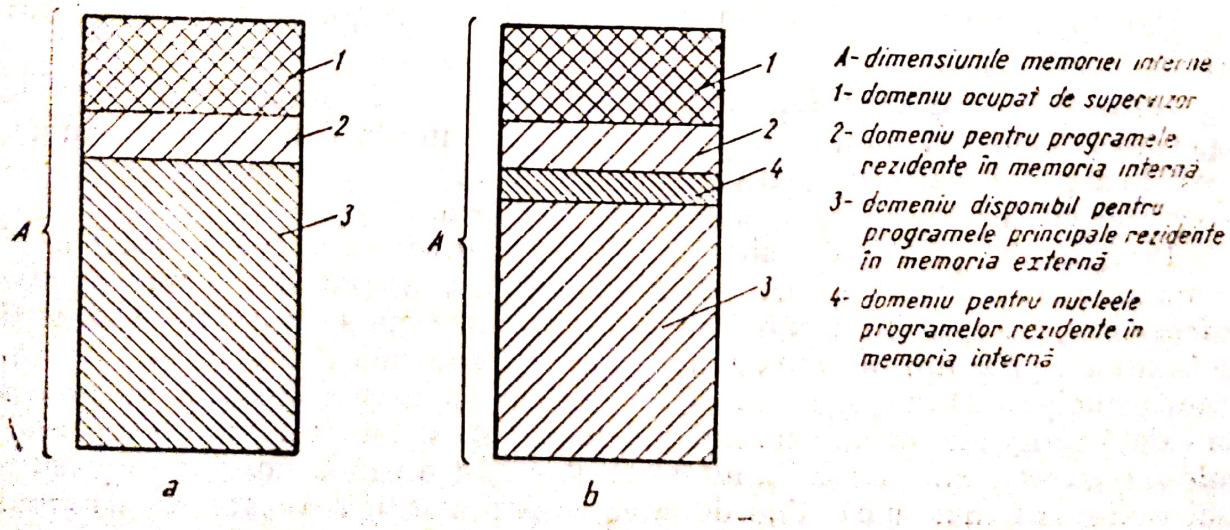


Fig. 12.15.

În celălalt caz, nucleul programului principal este rezident în memoria internă (fig. 12.15, *b*). În această situaţie este avantajos să se atribuie nucleului, în afara funcţiei de coordonare a programelor parţiale, şi alte funcţiuni, deoarece se obţin execuţii foarte rapide. Prin acest mod de organizare a programelor de aplicaţie este posibilă aplicarea multiprogramării pentru mai multe programe principale şi deci şi a mai multor programe parţiale.

Subprograme. Un procedeu suplimentar pentru alcătuirea programelor principale constă în tehnica subprogramelor. Subprogramele sînt utilizate, în primul rînd, pentru realizarea unor operaţii care se reiau de mai multe ori, în diferite locuri ale unui program. Exemple în acest sens: anumite expresii aritmetice complicate, efectuarea repetată a unor sume, determinarea timpului etc. Caracteristic pentru un subprogram este faptul că acesta există într-un singur exemplar, şi ori de cîte ori este utilizat într-un program principal, se face apel la acesta.

Subprogramul nu are un număr propriu ci este executat sub numărul programului principal care l-a apelat. Saltul de la programul principal la subprogram şi înapoi nu necesită o coordonare din partea sistemului de operare. Dacă se realizează alimentarea subprogramului cu parametrii necesari (datele de prelucrat), precum şi preluarea rezultatelor livrate de subprogram, folosirea tehnicii subprogramelor devine foarte simplă pentru utilizator.

Folosirea subprogramelor oferă o serie de avantaje (fig. 12.16). În locul amplasării aceleiaşi secvenţe de instrucţiuni în mai multe locuri (fig. 12.16, *a*), prin saltul repetat la secvenţa de instrucţiuni amplasată separat, distinct (fig. 12.16, *b*), se reduc dimensiunile programului. Un subprogram poate fi utilizat, de asemenea, de mai multe programe (fig. 12.16, *c*). În aceste cazuri poate apare situaţia în care un subprogram în curs de execuţie este solicitat de un alt program de prioritate mai mare. Pentru evitarea unor erori legate de acest aspect, se pot utiliza următoarele posibilităţi:

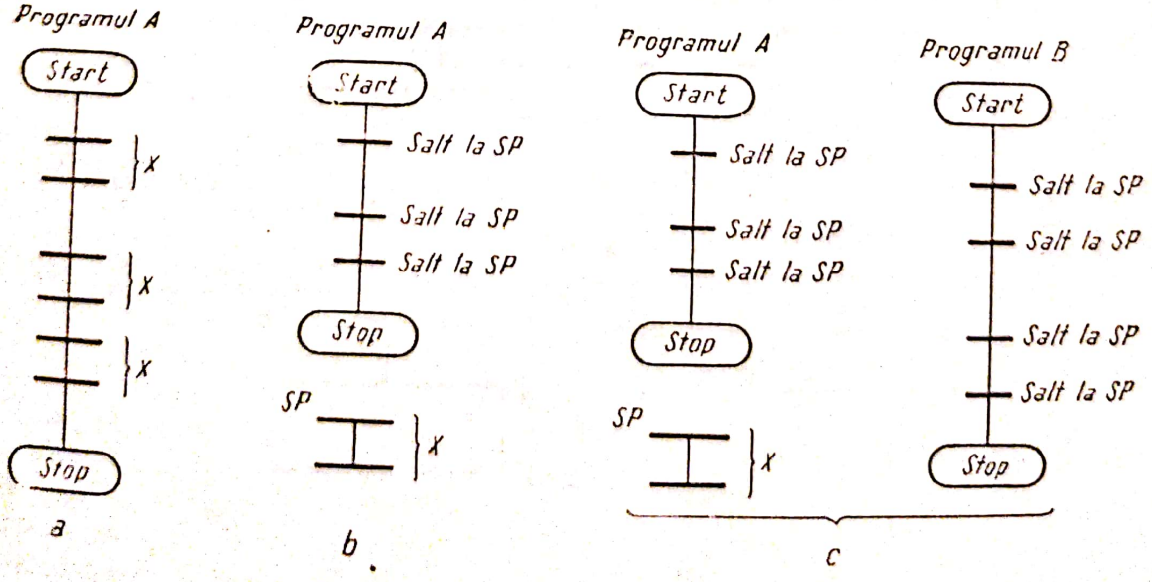


Fig. 12.16.

— subprogramul nu este întreruptibil, inclusiv de programe cu priorităţi superioare (desavantaj : contradicţie faţă de regula priorităţii) ;

— subprogramul este astfel scris încît partea instrucţiunilor de comandă să fie separată de domeniul cu date, aceasta permiţînd ca partea cu instrucţiunile de comandă să poată lucra în acelaşi timp cu mai multe seturi de date.

Un subprogram poate fi chemat de un program principal, de un program parţial sau de un subprogram.

Transferări de date. Deoarece un program principal este alcătuit din programe parţiale şi subprograme, pe de o parte, iar programele de aplicaţie conţin mai multe programe principale, pe de altă parte, în mod necesar apare problema transferării de date între acestea. Există diferite modalităţi de a asigura transferul de date între programe. Prezentăm în cele ce urmează o metodă utilizată curent.

În figura 12.17 este schiţată transferarea datelor cu ajutorul variabilelor ajutătoare între un program principal şi un subprogram. Numărul parametrilor transferaţi (datelor) depinde de fiecare caz concret în parte. Înainte de efectuarea saltului la subprogram, programul principal încarcă lista de variabile. Urmează apoi saltul la subprogram, după care variabilele din listă primesc valorile corespunzătoare. După cum se observă din figura 12.17, subprogramul conţine posibilităţile de prelucrare a datelor şi de încărcare a listei cu variabile pentru rezultate, prin intermediul cărora acestea ajung la programul principal.

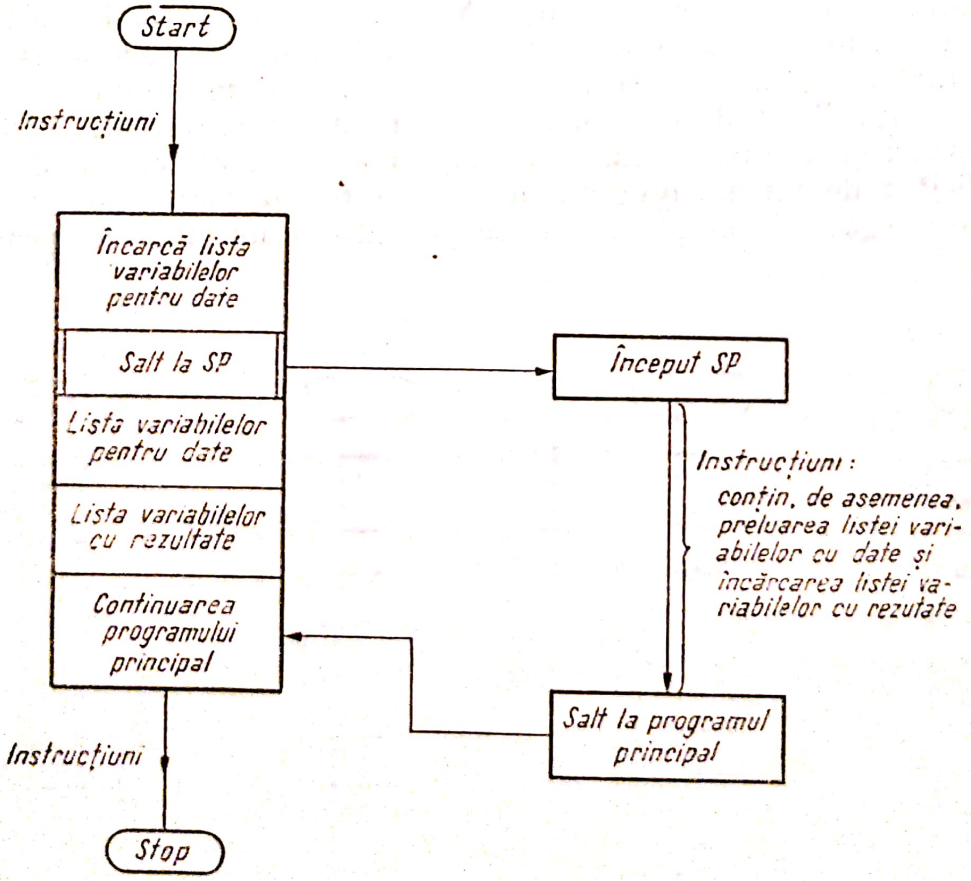


Fig. 12.17.

Scanned with OKEN Scanner

Atunci cînd programarea se face într-un limbaj de asamblare care poate prelucra macroinstrucțiuni, transferul de date între programe se poate face relativ ușor prin intermediul acestora [0].

Sisteme de programe de aplicație

Prin această noțiune se înțelege *ansamblul programelor* de aplicație necesare rezolvării problemelor concrete pentru care a fost implementat calculatorul de proces. La întocmirea sistemului de programe se ține seama de configurația echipamentului de calcul. Sistemul de programe presupune rezolvarea completă a problemei puse, chiar dacă este vorba de un calculator mic (cu o periferie minimă și numai un program principal) sau de un calculator mare, un calculator dublu sau calculatoare ierarhice cu o mulțime de programe principale independente sau dependente, programe parțiale sau subprograme sare au de prelucrat probleme de proces directe și/sau indirecte.

Punctele de vedere, care sînt hotărîtoare pentru alcătuirea unui sistem de programe, presupun cunoașterea anticipată a părților din care se compune un sistem de programe; acestea vor fi expuse în continuare.

Părțile componente ale unei sistem de programe. Problema de ansamblu, care se va rezolva cu sistemul de programe, trebuie divizată în problemele principale componente. Problema de ansamblu *„supravegherea cu calculatorul de proces a instalației de piroliză"* poate fi divizată, de exemplu, în următoarele probleme principale:

— prelucrarea intrărilor analogice;
— prelucrarea intrărilor binare și alarmelor;
— protocolarea;
— deservirea calculatorului;
— determinarea de caracteristici.

Ansamblul de programe necesare pentru rezolvarea uneia din aceste probleme principale se numește *bloc de programe*. Fiecare bloc de programe se compune la rîndul său, în funcție de problema pusă spre rezolvare, dintr-un număr mare de programe. Acestea se pot reduce la un program principal, alcătuit din mai multe programe parțiale și subprograme, căruia i se asociază un nume, de exemplu PRIAN — *PR*elucrarea *I*ntrărilor *AN*alogice — sau pot fi efectiv mai multe programe principale cu divizările corespunzătoare.

Divizarea problemei în părțile cele mai mici, care cuprind însă probleme bine definite, conduce la *programul de bază*.

Programul de bază nu mai poate fi divizat. Programe de bază tipice ale blocului de programe PRIAN pot fi, de exemplu:

— controlul mărimilor față de limite fixe;
— controlul mărimilor față de limite variabile;
— evoluția valorilor măsurate;
— liniarizarea.

Programele de bază stau la dispoziție sub formă de subprograme și/sau programe parțiale care rezolvă probleme bine definite. Prin combinarea programelor de bază corespunzătoare pot fi alcătuite blocuri de programe pentru rezolvarea problemelor de ansamblu specifice.

Scanned with OKEN Scanner

În cazul programelor de bază, asemănător cu situația întîlnită la subprograme și programe parțiale, apare problema transferului de date.

Zonele pentru transferul datelor pun la dispoziția programului de bază datele necesare prelucrării și preiau de la acesta rezultatele, respectiv datele necesare prelucrărilor în continuare. Aceste zone de transfer al datelor *decuplează* programele de bază, ceea ce înseamnă că între acestea *nu există schimburi directe* de informație. Dacă se observă mai atent datele necesare în execuția unui program, se ajunge la concluzia că acestea se pot împărți în trei grupuri (fig. 12.18):

— *date fixe*, care exprimă mărimile specifice ale procesului tehnologic (exemple: descrierea traductorului — analog sau digital, numărul punctului de măsurare, limitele fixe ale variabilei etc.);

— *date de cuplare* cele prelucrate de un program de bază și care sînt necesare ca parametri pentru un alt program de bază în cadrul blocului respectiv sau al unui alt bloc;

— *date variabile* cele care servesc ca date de control pe durata intervalului de prelucrare a unui program de bază.

Deoarece datele specifice unei probleme, în scopul unei manipulări ușoare, sînt continuu listate, s-a introdus și noțiunea *listă de date*. În acest sens, o zonă pentru transferul datelor se compune din date fixe, date de cuplare și date variabile. Pentru că în interiorul unui sistem de programare există mai multe liste pentru alimentarea cu date, acestea primesc nume specifice cum sînt: liste cu valori primare, liste cu valori măsurate, liste cu valori calculate etc.

Prin introducerea zonelor pentru transferul datelor este posibil ca, într-o mare măsură, programele de bază să fie independente de echipamentul de calcul. Parametrii necesari și specifici implementării curente sînt introduși în listele de aprovizionare cu date.

După cum s-a arătat, blocurile de programe specifice aplicației sînt alcătuite din programe de bază corespunzătoare. În continuare, la acestea se adaugă listele pentru ficeare în parte. Pentru execuția unui bloc de programe astfel alcătuit, este necesar încă un program de comandă care coordonează fiecare element de bază în parte.

Un bloc de programe alcătuit din programul de comandă, programele de bază și listele de date (fig. 12.18) poate fi modificat sau dezvoltat aproape arbitrar prin aceea că, păstrîndu-se structura blocului de programe, se înlocuiesc, se îndepărtează sau se adaugă programe de bază și liste.

Pentru utilizatorul unui bloc de programe este relativ simplu să dezvolte programe noi pe care să le cupleze prin intermediul zonelor de transfer al datelor. Un asemenea program suplimentar poate nu numai să preia informație din liste ci să și adauge. Prin aceasta este posibil să se transmită blocului de programe probleme noi spre rezolvare.

Sisteme de programe standard. Punerea spre rezolvare a unor probleme identice sau foarte asemănătoare conduce la observația că programele necesare ar putea fi standardizate. Aceasta înseamnă alcătuirea unei serii de programe de bază standard care să rezolve optimal părți distincte ale problemei respective. Standardizarea acestor programe necesită, printre altele, precizarea numărului, construcției și felului listelor. În măsura în care pentru aplicația concretă se cunosc exact cerințele privind configurația

Scanned with OKEN Scanner

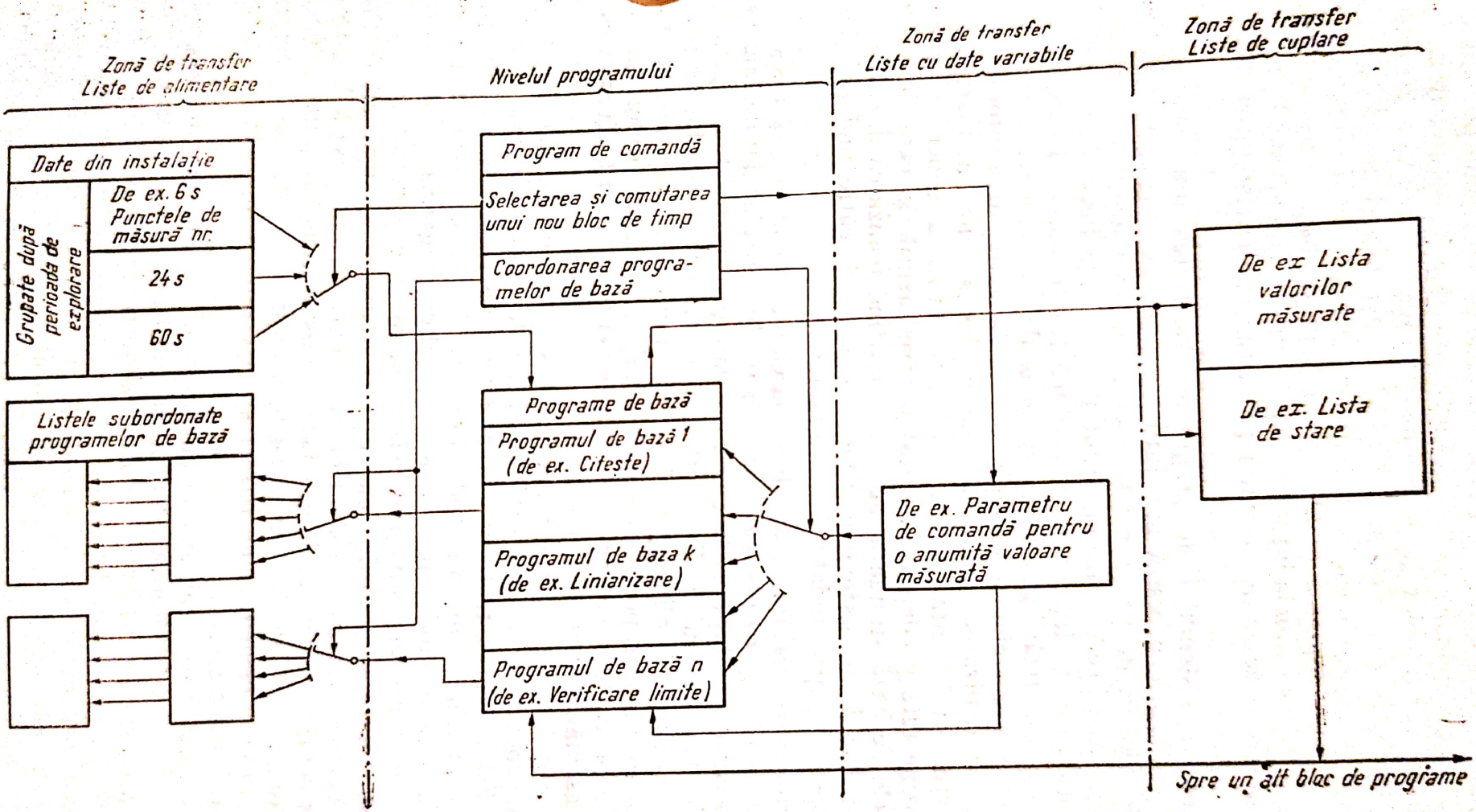


Fig. 12.18.

Scanned with OKEN Scanner

calculatorului, programele de bază, blocurile de programe precum și timpul real, în aceeași măsură pot fi definite programele de comandă.

Utilizatea sistemelor de programare standard prezintă mai multe avantaje: programele de bază sînt bine documentate și temeinic întocmite, performanțele acestora sînt cunoscute și verificate prin folosirea repetată. Pentru aplicația concretă se aleg programele de bază corespunzătoare, cu acestea trecîndu-se apoi la întocmirea sistemului de programe. Prin acest mod de lucru se reduce substanțial timpul pentru alcătuirea sistemului de program.

Generarea sistemului de programe. Alcătuirea unui sistem de programe de aplicație necesită mai întîi compunerea programelor de bază și listelor în blocuri de programe și apoi înlănțuirea acestora. Această activitate este laborioasă, consumă mult timp și este însoțită de greșeli care în majoritate pot fi detectate numai după testări ample. De aceea s-a pus problema automatizării generării sistemelor de programe, care, deși necesită introducerea unor elemente suplimentare la întocmirea programelor de bază și chiar realizarea unor programe în plus, s-a dovedit foarte eficientă [0]. Efortul suplimentar solicitat de automatizarea generării sistemelor de programe este pe deplin justificat pentru cazurile în care sînt utilizate programele de bază standard, acestea permițînd realizarea diferitelor combinații, și deci de noi sisteme de programe, chiar de către inginerii chimiști fără cunoștințe aprofundate de programare.

Primul și cel mai important lucru la elaborarea automată a sistemelor de programe este *generatorul de liste*. Pentru obținerea listelor de alimentare, datele specifice instalației se trec pe formulare și de pe acestea se perforează pe cartele. Generatorul de liste sortează aceste date pe probleme parțiale și le trece în listele de alimentare. Listele deținute în acest mod sînt trecute în continuare pe un suport fizic și sînt la dispoziția utilizatorului. La citirea listelor de pe cartele, generatorul de liste verifică corectitudinea acestora în mod asemănător unui compilator.

Următorul pas pentru automatizare este *alcătuirea sistemelor de programe* din programele de bază și din listele de alimentare realizate cu generatorul de liste. Pentru realizarea acestui pas există diferite modalități, în continuare prezentîndu-se o metodă care are la bază un *program de generare*. Această metodă este independentă de limbajul de programare și de compilator, fapt ce îi artibuie un grad relativ mare de generalitate [0].

Lucrul cu programul de generare presupune că programul de bază și listele necesare alcătuirii sistemelor de programe sînt puse sub o formă de înțeles pentru programul de generare. Această cerință impune ca fiecare program de bază să fie prezentat într-o formă aprioric fixată, care precizează condițiile în care elementul de bază respectiv este introdus la generare în sistemul de programe.

Pentru alcătuirea unui sistem de programe este necesar un ansamblu de elemente de bază. Ordinea elementelor de bază în acest ansamblu este dată prin procesul de prelucrare al sistemului de programe ce trebuie generat. Ansamblul elementelor de bază poate fi depus pe un suport de memorie externă sau într-un pachet de cartele. Listele produse anterior de generatorul de liste, alcătuite astfel încît să respecte condițiile de includere în sistemul de programe, sînt fie incluse în ansamblul elementelor de bază, fie depuse într-un bloc de date separat.

Scanned with OKEN Scanner

Pentru generarea unui sistem de programe, utilizatorul comunică programului de generare procesul de prelucrare sub forma unei liste de „dorințe". Corespunzător acestei liste de „dorințe", programul de generare trece în revistă toate elementele de bază existente în ordinea fixată și pe baza condițiilor impuse verifică dacă un anumit element de bază trebuie sau nu introdus în sistemul de programe ce se generează. Programele de bază ce trebuie incluse sînt supuse unui proces de translatare astfel încît atunci cînd ultimul element de bază al ansamblului a fost introdus, sistemul de programe dorit este gata.

Printre altele, metoda prezentată are avantajul că permite inginerilor chimiști, neantrenați în programare, să elaboreze sisteme de programe.

Așa cum se arată în paragraful 12.3, după o perioadă relativ lungă de investiții, însoțită de eforturi materiale apreciabile, în ultimul timp au fost puse la punct *limbaje universale de programare în timp real*, a căror aplicare va îmbunătăți substanțial activitatea de realizare a sistemelor de programe.

12.3. LIMBAJE DE PROGRAMARE PENTRU CALCULATOARE DE PROCES

Spre deosebire de limbajele utilizate la programarea calculatoarelor universale, limbajele de programare pentru calculatoarele de proces prezintă unele trăsături specifice, determinate de problemele ce se cer a fi rezolvate cu acestea. Astfel, pentru un calculator de proces se pun spre rezolvare următoarele tipuri de probleme [0]:

1. supravegherea proceselor;
2. reglarea numerică a proceselor;
3. comanda proceselor secvențiale;
4. prelucrarea datelor din proces în vederea determinării modelelor matematice ale proceselor;
5. simularea proceselor;
6. probleme cu specific economic (cazul optimizării);
7. tratarea unor probleme științifice;
8. lucrări specifice activității unui centru de calcul.

Multitudinea laturilor acestor probleme îngreunează foarte mult alcătuirea unui limbaj de programare pentru calculatoarele de proces. Problemele de tipul 7 și parțial 8 pot fi acoperite în bună măsură cu ajutorul limbajelor FORTRAN și ALGOL, la care trebuie adăugate proceduri în cod-mașină pentru proces. Pentru problemele de tipul 6 poate fi utilizat limbajul COBOL. Drept aplicații specifice proceselor sînt problemele de tip 1; 2; 3; 4 și 5, pentru a căror programare optimă este necesar un limbaj de programare corespunzător calculatoarelor de proces. În continuare din motive de exprimare sintetică, se va utiliza denumirea *limbaj de proces*.

Pentru a evidenția aceste aspecte este util de prezentat un exemplu din lucrarea [0]. Una din aplicațiile curente ale unui calculator de proces constă în prelucrarea automată a cromatogramelor în timp real. Pentru prelucrarea în sine a cromatogramei, chiar și în cazul peak-urilor puternic suprapuse, a fost pus la punct un program în ALGOL mult apreciat de utilizatori din punctul de vedere al preciziei și reproductibilității rezultatelor.

Scanned with OKEN Scanner

Din punctul de vedere al programatorului de proces, trebuie depus în continuare un volum mare de muncă pentru întocmirea programelor de aplicație într-un limbaj de asamblare care să realizeze:

a. colectarea datelor, simultan de la mai multe cromatografe;

b. prezentarea datelor măsurate spre prelucrare cu programul scris în ALGOL;

c. utilizarea datelor obținute din prelucrarea cromatogramei în sensul aplicației respective.

Întocmirea programelor pentru efectuarea acestor operații necesită un efort care este cel puțin egal cu cel depus la realizarea programului în ALGOL pentru prelucrarea cromatogramei.

În cazul utilizării unui limbaj de proces pentru această problemă, *a* se rezolvă prin instrucțiunile de introducere standard conținute, iar *b* nu mai este necesară deoarece programarea prin limbajul de proces pune la dispoziție datele automat în forma necesară. În această situație este necesar a se întocmi programe în limbajul de proces numai pentru problema *c*.

Cerințele impuse unui limbaj de proces

În cele ce urmează se vor prezenta principalele cerințe care se impun unui limbaj de proces:

1. Să facă față *desfășurării simultane* a diferitelor părți ale procesului. Aceasta înseamnă că trebuie să prezinte facilități pentru modul de lucru în timp real, care, pe baza priorităților asociate fiecărei activități, satisface cerințele de simultaneitate impuse de proces.

2. Să permită *realizarea fără dificultăți a intrărilor/ieșirilor* în/din sistem, care, în cazul calculatoarelor de proces prezintă două aspecte: unele au loc în mod regulat la intervale de timp fixe iar altele, cum este cazul alarmelor, la intervale aleatoare. Datorită dezvoltării vizualizării procesului cu ajutorul tuburilor cinescop, limbajul de proces trebuie să conțină instrucțiuni corespunzătoare pentru extragerea informației.

3. Să permită *programarea* atît a *proceselor continui* cît și a celor *secvențiale*.

4. Cu toată tendința de mărire și de ieftinire a memoriei, este necesar să se acorde atenție posibilității de *segmentare a programelor*. În acest sens nu trebuie neglijate facilitățile de segmentare a datelor sau mai general, a transferării de date de la/spre suporturile externe de date.

5. Să conțină elemente pentru realizarea eficientă a *testărilor*.

Realizarea unor limbaje de programare care să îndeplinească toate aceste cerințe este o sarcină foarte complexă. Numai așa se explică faptul că, spre deosebire de domeniul limbajelor de programare pentru problemele științifice (ALGOL și FORTRAN) și economice (COBOL), în domeniul limbajelor de proces se constată o rămînere substanțială în urmă.

În ultimul timp, printr-o conlucrare largă și susținută între diferiții producători de echipamente de calcul, utilizatori și cercetători, s-au obținut rezultate îmbucurătoare în acest sens. Așa este cazul limbajului de proces PEARL elaborat în R. F. Germania*.

---

* Martin, T. *Zur Situation der PEARL-Entwicklung in BRD*. Regelungstechnische Praxis, 18 (1976), Heft 7.

Limbaje de programare utilizate

Pînă în prezent, în domeniul calculatoarelor de proces s-au folosit, sau sînt în curs de folosire, categoriile de limbaje prezentate shematic în figura 12.19. Din această figură rezultă că există *limbaje dependente de calculator* și *limbaje independente de calculator*.

Limbajele din prima categorie prezintă următoarele dezavantaje:

— sînt depărtate de proces;

— necesită un efort foarte mare pentru a fi învățate;

— au o viață scurtă corespunzătoare vieții echipamentului pentru care au fost alcătuite;

— conduc la prețuri de programare apreciabile;

— fiind dependente de echipament rezultă că numărul acestora este la fel de mare ca cel al fabricanților de calculatoare.

Limbajele din categoria a doua, independente de calculator, elimină în mare parte dezavantajele limbajelor dependente de calculator. La rîndul lor, acestea se divid în:

— limbaje specifice aplicației, orientate pe problemă;

— limbaje universale, orientate pe procedură.

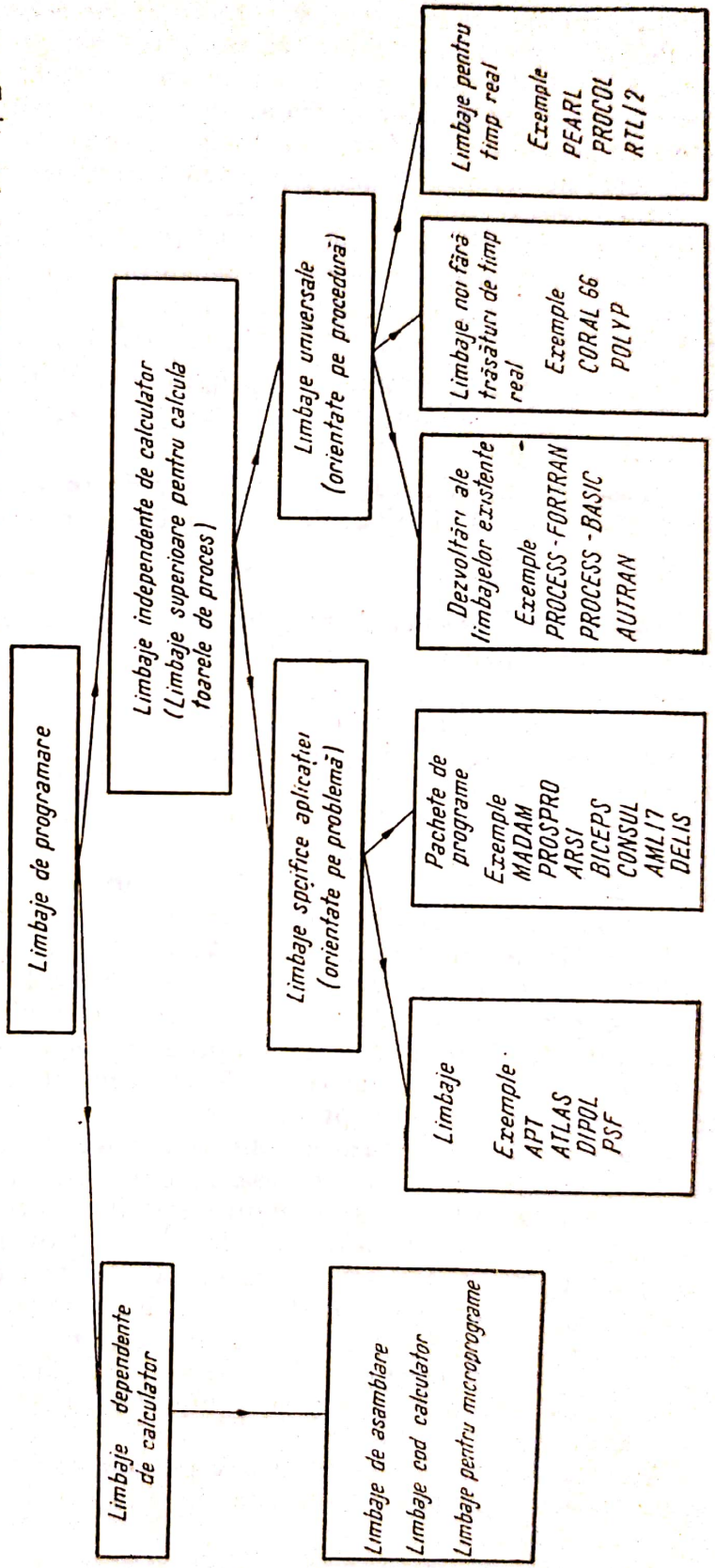


Fig. 12.19.

Scanned with OKEN Scanner

Limbajele specifice aplicației. Aceste limbaje sînt divizate, de asemenea, în două grupuri: *limbaje pentru aplicații cu trăsături bine precizate*, cum sînt, de exemplu, EXAPT pentru programarea problemelor din construcția de mașini, ATLAS pentru testarea aparaturii electronice sau DIPOL pentru automatizarea proceselor secvențiale; în cealaltă grupă sînt conținute *sisteme de programe sub formă de pachete pentru probleme speciale*. De obicei, aceste sisteme de programe utilizează formulare tip care sînt completate (fill in the blanks) conform unor indicații sumare foarte apropiate de noțiunile familiare specialiștilor. După completarea acestora, conținutul este trecut pe cartele prin a căror citire se generează programele de conducere corespunzătoare. Din această categorie, la noi în țară sînt utilizate sistemele PROS PRO, produs al firmei IBM—SUA, și DELIS, elaborat de IPA-România.

Deoarece aceste sisteme de programe pot fi dezvoltate prin mijloace asemănătoare limbajelor, s-a convenit să se atribuie și acestora caracterul de limbaj de programare.

Lipsa de maleabilitate a sistemelor de programe din această grupă — — posedă un număr limitat de alternative — reprezintă un mare dezavantaj al acestora.

Limbaje universale. Aceste limbaje sînt denumite universale deoarece sînt aplicabile pentru un spectru larg de aplicații ale calculatoarelor de proces.

Un asemenea limbaj conține un set de instrucțiuni care pune la dispoziția programatorului posibilități practic nelimitate de programare, performanțele programului de conducere depinzînd numai de capacitatea celui care l-a întocmit. Pentru punerea în valoare a unui asemenea limbaj, analistul trebuie să cunoască foarte bine atît procesul tehnologic cît și posibilitățile sistemului de calcul și conducere automată.

Limbajele universale sînt divizate în trei grupe:

Dezvoltări ale limbajelor existente. Unele limbaje universale, de exemplu FORTRAN, au fost dezvoltate în sensul de a fi utilizabile și pentru programarea problemelor de proces. Aceste dezvoltări sînt de fapt numai instrucțiuni de chemare a unor subprograme speciale care sînt greu de parametrizat. Prin aceasta, limbajele de bază nu au devenit totuși apte pentru programarea eficientă a proceselor.

Limbaje noi fără trăsături de timp real. A existat tendința ca pentru limbajele create în ultima perioadă, cum sînt de exemplu CORAL 66 și POLYP, să se adauge suplimente specifice programării problemelor de proces. Deoarece aceste adăugiri depind de proveniența fiecărui calculator, limbajele de acest tip nu au cunoscut decît o arie restrînsă de utilizări.

Limbaje pentru timp real. Aceste limbaje au fost dezvoltate pe baza experienței acumulate în domeniul programării cu limbajele prezentate mai sus, și pe baza cerințelor impuse calculatoarelor de proces. Un exemplu de asemenea realizare îl constituie limbajul PEARL (*Proces and Experiment Automation Realtime Language*) dezvoltat de curînd în R. F. Germania*.

Limbajele din această grupă posedă calitatea efectivă de a fi concepute special pentru activitățile în timp real. Aceasta înseamnă că ele

---

* Martin, T. *Höhere Programmiersprachen für Prozeßbrechner*. Regelungstechnische Praxis, 18 (1976), Heft 9.

Scanned with OKEN Scanner

dispun de posibilităţi care permit transpunerea fără dificultăţi a proceselor tehnice ce se desfăşoară în timp real în procesele de calcul dintr-un calculator de proces.

Exemple

În continuare vor fi prezentate, spre exemplificare, unele elemente privitoare la anumite limbaje de proces, precum şi unele programe alcătuite cu ajutorul acestora.

Ex. 12—4. **Limbajul PROSPRO.** Creat de firma IBM, acest limbaj este bazat pe completarea unor formulare (fill in the blanks) care necesită din partea utilizatorului cunoştinţe temeinice despre proces şi un minim de cunoştinţe despre limbajul propriu-zis. Denumirea sa reprezintă abrevierea expresiei *PROces Systems PROgrams*.

PROSPRO permite lucrul în timp real, calculatorul fiind conectat la proces. Principalele operaţii care se realizează cu ajutorul acestui limbaj sînt:

— explorarea variabilelor din proces şi verificarea limitelor acestora periodic, la intervale de 40 ms;

— prelucrarea periodică a variabilelor (perioadele pot fi modificate în intervalul 1...90 minute);

— modificarea prescrierilor regulatoarelor în cazul conducerii supervizate a procesului;

— reglarea numerică directă (DDC), operaţie posibilă numai în cazul variantei PROSPRO 2.

Informaţiile privind strategia supravegherii şi conducerii procesului sînt transmise calculatorului prin completarea unui număr de formulare sum sînt cele care conţin:

— variabilele ce trebuie prelucrate;
— informaţii asupra variabilei;
— relaţia generală;
— informaţii de ajustare;
— operaţii generale etc.

Unele mărimi, cum este cazul debitului la gaze, sînt obţinute în urma efectuării anumitor calcule. Limbajul PROSPRO are prevăzută în acest scop o *formă generală de relaţie*, şi anume:

$$X = A + B(C + (F/D))^E, \qquad (12-1)$$

unde A, B, C, D, E şi F sînt mărimi primare din proces sau constante pentru care sînt precizate valorile corespunzătoare. Pentru fiecare caz concret, aceste mărimi sînt egale cu 0; 1 sau iau valori curente din proces.

Sistemul PROSPRO este prevăzut şi cu posibilitatea ajustării unor mărimi pe baza unei *ecuaţii de corecţie*, scrisă cu ajutorul unor creşteri finite:

$$F_1\Delta T + F_2\Delta M + F_2\Delta U + F_4\Delta V_1 + F_5\Delta V_2 + F_6\Delta V_3 = 0 \qquad (12-2)$$

unde: $F_1, F_2, \ldots, F_6$ sînt coeficienţi;
$\Delta V_1, \Delta V_2, \Delta V_3, \Delta U$ — variaţii ale perturbaţiilor;
ΔM este comanda;
ΔT — variabila dependentă.

Scanned with OKEN Scanner

Ecuația de corecție trebuie înțeleasă astfel: atunci cînd se modifică perturbațiile cu valorile ΔV_1, ΔV_2, ΔV_3 și ΔU, cu cît trebuie modificată comanda M, astfel încît variabla dependentă T să varieze cu ΔT impus.

Formularul pentru relația generală și formularul cu informații de ajustare conțin datele necesare pentru utilizarea relației generale (12—1) și a ecuației de ajustare (12—2) pentru fiecare caz concret în parte.

Formularul cu informații asupra variabilei. În continuare se prezintă mai îndeaproape formularul cu informații asupra variabilei. Informațiile conținute de acest formular sînt grupate în șase categorii:

liniile 00...05, denumite și intrări, conțin informații generale;
intrările 10...18 conțin informații pentru determinarea valorii mărimii respective;
intrarea 19 indică o acțiune specială intermediară;
intrările 20...35 conțin informații asupra acțiunii speciale și de verificare a limitelor;
intrările 40...50 conțin informații asupra acțiunii de modificare a prescrierii;
intrarea 51 indică o acțiune specială finală.

Conținutul acestui formular este transpus pe cartele; fiecare intrare, deci fiecare linie, generează o cartelă perforată după cum urmează:

coloana 1 conține un zero, asociat cartelelor pentru formularul cu informații asupra variabilei;
coloanele 2...5 conțin numărul asociat variabilei;
coloana 6 conține un blank;
coloanele 7 și 8 conțin codul asociat intrării de pe formular;
coloanele 10...25 conțin informațiile asupra variabilei.

Prezentarea conținutului intrărilor (tabelul 12.1):

- Intarea *00* conține un text alfanumeric de 16 caractere care servește la identificarea variabilei. În cazul de față, acest text este F704 B/FRC—201 C1.
- Intrarea *01* specifică unităților de măsură ale variabilei. care, pentru variabila de față sînt m^3/h.
- Intrarea *02* indică secvența de prelucrare pentru determinarea valorilor anterioare utilizate pentru determinarea valorii variabilei în cazul pornirii sistemului.
- Intrarea *03* precizează intervalul de timp în minute la care se prelucrează variabila.
- Intrarea *04* indică ponderea valorii curente a variabilei în stabilirea valorii medii a acesteia. Relația utilizată pentru calculul valorii medii este

$$X_{mi} = \frac{(n-1)X_{m,i-1} + X_i}{n}, \qquad (12-3)$$

unde: $X_{m,i-1}$ este valoarea medie anterioară;
X_{mi} — valoarea medie nouă;
X_i — valoarea curentă;
n — conținutul intrării *04*.

Dacă intrarea *04* conține 0 sau blanc, nu se calculează valoarea medie.

Scanned with OKEN Scanner

IBM PROSPRO/1800

Tabelul 12.1

1	5
0	,704

Informatii asupra variabilei (Variable Information)

Denumirea DEBITUL DE MATERIE PRIMA PENTRU REACTORUL 1

FRC - 201 C1

		7 8 10	
*	00	F,7,0,4,3,/,F,R,C,-,2,0,1,,C,1	Simbol de identificare
*	01	N,.,.,5,/,H,	Unitati de masura
*	02	0,0,2,0	Secvente de prelucrare
	03	0,3	Interval de prelucrare
	04	0,2	Numar.valorilor pentru calculul valorii medii
	05		Se utiliz.media pt.verif.refer.variat.curente si abaterii? (1 = Da)
	10	1	Prelucrarea valorii curente
	11	1	Masurare directa? (1 = Da)
	12	0,2,5,1	Supraveghere continua pt.max.-min.? (1 = Da)
	13	,,3,2,7,6	Adresa marimii masurate
	14	1,3,1,0,7	Indicatia CAN la limita inferioara a scalei (a)
	15	1	Domeniul de masura in unitati CAN (b)
	16	0,.,0	Optiune de radical in ec.de conversie (1 = Da)
	17	3,0,0,0,.,0,	Limita inferioara a domeniului de masura (B)
	18	1,0,1,9,2	Domeniul de masura al aparatului (C)
	19	1,0,1,8,0	Calculul special al valorii curente
			Actiune speciala intermediara
			Verificarea limitelor
*	20	2,8,0,0,.,0,	Limita maxima
	21		Normal→Depasit / Depasit-Normal: Actiune speciala cind se depaseste limita maxima
	22		
*	23	5,0,0,.,0	Limita minima
	24		Normal→Depasit / Depasit-Normal: Actiune speciala cind se depaseste limita minimă
	25		
	26		Valoare limita sup.a zonei de operare
	27		Dedesubt→Deasupra / Deasupra→Dedesubt: Actiune speciala cind se depaseste aceasta valoare
	28		
	29		Valoare limita inf.a zonei de operare
	30		Dedesubt = Deasupra / Deasupra = Dedesubt: Actiune speciala cind se depaseste aceasta valoare
	31		
	32		Variatie curenta max.admisa pt.semnalizare
	33		Act.spec.cind variat.curentă admisă este depașită
	34		Variatia curenta necesara pt.o corect.predictiva
	35		Actiunea pentru o corectie predictiva
	40		Prelucr.valorii prescrise (target) si a abaterii
	41		Are variab.o valoare prescrisa? (target) (1= Da)
	42		Timp.min.intre doua calc.ale val.pres.(target) si/sau corectie a abaterii
	43		Actiunea de calcul a noii val.prescrise (target)
	44		Abaterea limita
	45		Actiunea cind se depaseste abaterea limita
	46		Abaterea pt.actiunea de corectie cu react.inversa
	47		Actiune pt.corectia normala a abaterii
	48		Corectia max.a marimii de refer.(set point) per pas
	49		Iesirea val.de refer.(set point) (1 regulator 2 mesaj
	50		Adresa regulatorului
	51		Viteza de variatie a marimii de refer-(set point) (1 = vit.max.)
			Actiune speciala finala

*Se completeaza obligatoriu.

Scanned with OKEN Scanner

Dacă $n = 2$, relaţia (12—3) devine

$$X_{mi} = \frac{X_{m,i-1} + X}{2},$$

ceea ce arată că noua valoare intră cu o pondere de 50% în calculul mediei curente.

Relaţia (12—3) este foarte utilă pentru calculul valorilor acelor variabile la care măsurarea se face în prezenţa unor semnale parazite puternice, situaţie specifică debitelor de lichid.

● Intrarea *05* precizează, cazurile în care valoarea medie este utilizată pentru a substitui valoarea curentă în procesul de verificare a depăşirii limitelor.

● Intrarea *10* indică dacă valoarea curentă a variabilei este determinată prin conversia unei mărimi de intrare măsurate (semnal electric). Dacă această intrare conţine un blank, valoarea curentă este calculată cu ajutorul unei relaţii ce utilizează alte variabile (cazul mărimilor derivate); în această situaţie, intrările 11...17 sînt ignorate.

● Intrarea *11* precizează dacă variabila este verificată continuu în raport cu limitele de minim şi maxim. Ori de cîte ori este depăşită una dintre limite, sistemul apelează o subrutină de alarmă.

● Intrarea *12* indică adresa mărimii măsurate.

● Intrările *13* ... *17* sînt utilizate pentru specificarea relaţiei de conversie de către convertorul analog-numeric.

● Intrarea *18* serveşte la precizarea modului în care se face corecţia semnalului cu relaţia generală (12.1) sau, în cazul temperaturilor, indică subrutina ce trebuie apelată pentru conversia semnalelor traductoarelor de temperatură.

● Intrarea *19* este utilizată pentru specificarea unei acţiuni intermediare asupra noii valori disponibile pentru variabila în cauză.

În continuare, pentru înţelegerea intrărilor *20*...*25* în figura 12.20 se prezintă tipurile de limite care sînt luate în consideraţie de calculator. Unei mărimi oarecare X îi sînt asociate următoarele valori:

— valoarea curentă;
— valoarea dorită;
— limitele inferioară şi superioară;
— limitele minimă şi maximă.

Valoarea dorită nu implică neapărat prezenţa reglării automate a mărimii X; pentru oricare variabilă există o anumită valoarea care prezintă anumite avantaje din punctul de vedere al procesului în cauză.

Valoarea curentă poate depăşi într-un sens sau altul limitele inferioară sau superioară de operare. Optional, prin intrările 27/28 respectiv 30/31 se poate preciza iniţierea unei acţiuni speciale. Atunci cînd valoarea curentă depăşeşte limitele minimă sau maximă, operatorul este avizat prin semnalizare şi printr-un mesaj scris. Intrările 21/22 şi 24/25 pot fi utilizate pentru specificarea unei acţiuni speciale.

● Intrările *20* şi *23* sînt utilizate pentru specificarea limitelor minimă şi maximă.

● Intrările *26* şi *29* indică limitele de operare inferioară şi superioară; utilizarea lor este opţională. Atunci cînd nu sînt utilizate, acestea se completează cu blank.

Scanned with OKEN Scanner

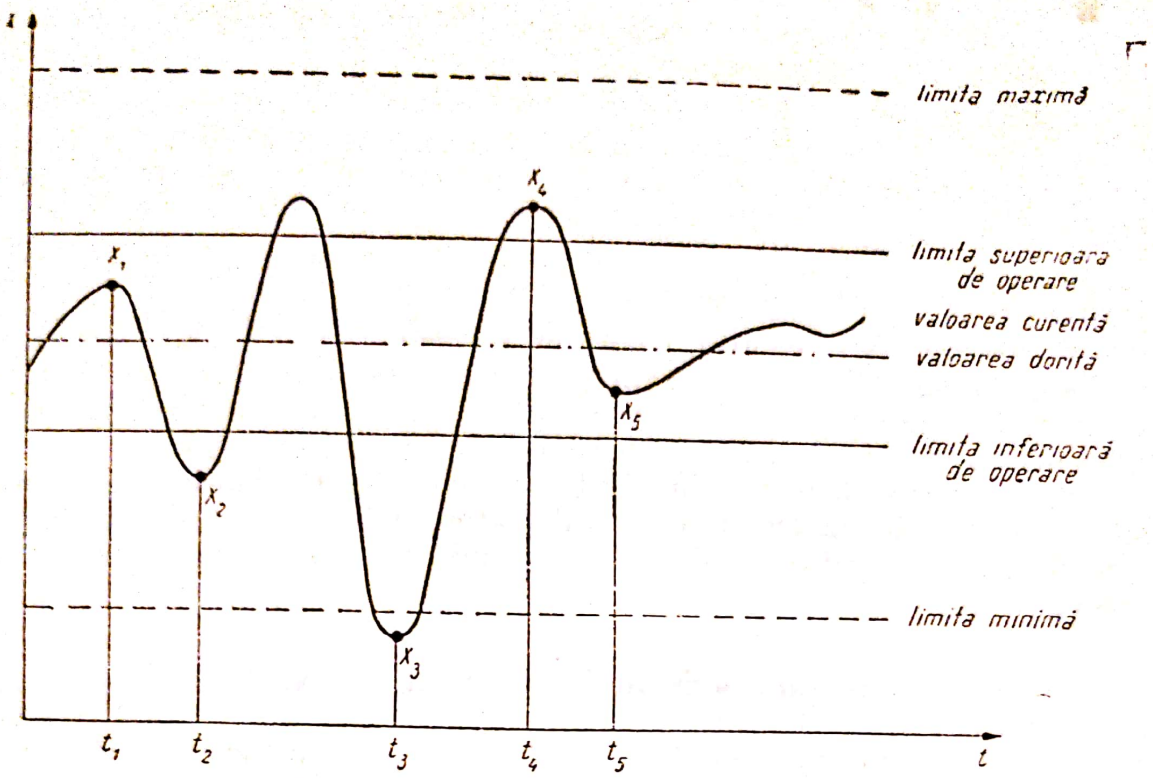


Fig. 12.20.

● Intrările *21* ; *22* ; *24* ; *25* ; *27* ; *28* ; *30* și *31* sînt utilizate pentru a indica acțiunea specială la traversarea limitelor de operare sau a limitelor minimă și maximă. Aceste intrări se completează cu blank cînd atunci acțiunea specială nu este dorită.

● Intrările *32* și *34* servesc pentru precizarea variației curente maxime admise pentru semnalizare și a variației curent necesare pentru o corecție predictivă.

● Intrările *33* și *35* sînt utilizate pentru indicarea acțiunilor speciale în cazul cînd variația curentă admisă este depășită respectiv pentru corecția predictivă.

Următoarele 11 intrări sînt asociate mărimilor care au valori prescrise.

● Intrarea *40* primește valoarea 1 atunci cînd pentru mărimea respectivă există o valoare prescrisă. În caz contrar, intrarea se completează cu blank, urmînd ca intrările *41* ... *50* să fie ignorate.

● Intrarea *41* specifică timpul minim dintre două calcule ale noii valori prescrise și ale abaterii în vederea corectării acesteia.

● Intrarea *42* precizează dacă se calculează valori noi pentru prescriere. Aceste valori noi se pot calcula cu relația generală (12.1) sau cu un program special, destinat acestui scop.

● Intrarea *43* indică limita absolută a abaterii.

● Intrarea *44* precizează dacă se apelează sau nu la o acțiune specială.

● Intrarea *45* arată valoarea absolută a abaterii pentru care se inițiază acțiunea de reglare, bineînțeles la intervalele de timp minim indicate prin intrarea *41*.

Scanned with OKEN Scanner

• Intrarea *46* specifică acţiunea ce trebuie întreprinsă pentru eliminarea abaterii.

• Intrarea *47* indică corecţia maximă pe pas a valorii prescrise.

• Intrarea *48* precizează modul în care va fi reglată valoarea de referinţă şi anume, direct de către calculator sau indirect de către operator, pe baza mesajului tipărit de calculator.

• Intrarea *49* indică adresa regulatorului pentru care valoarea prescrisă se fixează prin intermediul calculatorului.

• Intrarea *50* specifică viteza de variaţie a valorii de referinţă.

• Intrarea *51* este utilizată pentru indicarea oricărei acţiuni suplimentare înainte ca procesorul de variabile să cedeze comanda calculatorului pentru alegerea următoarei variabile ce trebuie prelucrată.

Fila prezentată în tabelul 12.1 este completată corespunzător unui anumit studiu de implementare a calculatorului de proces şi anume culegere de date şi verificarea limitelor minime şi maxime ale mărimilor din proces.

Ex. 12—5. **Program scris în limbajul AUTRAN.** Acest limbaj a fost dezvoltat de firma CDC pentru programarea problemelor privind conducerea proceselor continui. Aşa după cum se observă din figura 12.19, acesta este un limbaj universal, orientat pe procedură. În cele ce urmează se prezintă unele instrucţiuni din acest limbaj de proces prin intermediul unui program destinat reglării automate a unei temperaturi* .

```
1 ANALOG SENSOR (T 012, 'TEMP.REACT. 1', 2)
2 CONVERT TO (200.0, 320.0, GRC) USING (RTD)
3. PERIOD (1.0)
4 CONTROLLER (R 012, 'REGL.TEMP.REACT.1', PIDE) I, V,
  INCREMENTAL
5 RANGE (0.0, 100.0, PNT)
6 FEED BACK (T 012)
7 SET POINT (R 011)
8 TUN ING (PIDE, 3.2, 4.5, 0.8, 1)
9 OUTPUT TO (CV 012)
```

Prima instrucţiune specifică datele necesare în vederea culegerii şi prelucrării unei intrări analogice. În exemplul de faţă este cazul unei temperaturi din reactorul 1, cu eticheta de identificare T 012, conectată la adresa fizică 2 a canalului multiplexorului.

Instrucţiunea 2 indică domeniul de măsură al mărimii analogice, în exemplul dat 200...320°C, şi subrutina utilizată pentru convertirea semnalului, care în cazul de faţă are numele RTD.

---

* Pirvan, N. *Limbaje de programare specializate pentru aplicaţii industriale.* Centrul de prelucrare automată a datelor şi consultanţă, Bucureşti, 1975.

Scanned with OKEN Scanner

În continuare, se indică perioada de testare ciclică a temperaturii, 1 secundă, și date asupra algoritmului de reglare, care este de tipul PID. Bucla de reglare are asociat numărul 012, prescrierea provine din comanda buclei 011 — instrucțiunea SET POINT (R 011) — iar mărimea de reacție de la traductorul de temperatură T 012 — instrucțiunea FEED BACK (T 012). Este deci cazul unei reglări în cascadă. Coeficienții algoritmului de reglare au valorile 3,2 pentru partea proporțională, 3,2 minute pentru partea integratoare și 0,8 minute pentru partea derivatoare.

Comanda buclei de reglare a temperaturii în reactor notată cu 012 acționează robinetul de reglare CV 012 — instrucțiunea OUTPUT TO (CV 012).

Scanned with OKEN Scanner

BIBLIOGRAFIE

0. Anke, K., Kaltenecker, H., Oetker, R. *Prozessrechner—Wirkungsweise und Einsatz.* Berlin, Akademie—Verlag, 1971.
1. Baltac, V. *Optimizarea sistemelor de operare ale calculatoarelor numerice.* Timişoara, Editura Facla, 1974.
2. Bolliet, L. *Utilisation des ordinateurs à distance en temps réel et en temps partagé.* Paris, Dunod, 1967.
3. Boudarel, R., Delmas, J., Guichet, P. *Commande optimal des processus,* tome 1. Paris, Dunod, 1967.
4. Costache, N. *Alegerea sistemelor electronice numerice de prelucrare a informaţiei.* Bucureşti, Editura tehnică, 1970.
5. Crowe, C. M. *Chemical Plant Simulation. An Introduction to Computer-Aided Steady-State Process Analysis.* Englewood Cliffs, Prentice Hall, 1971.
6. Desmonde, W. H. *Computers and Their Uses.* Englewood Cliffs, Prentice-Hall, 1969.
7. Desmonde, W. H. *Real-Time Processing Systems: Introductory Concepts.* Englewood Cliffs, Prentice-Hall, 1964.
8. Dumitrescu, I., Florovici, M., Macri, I., Marinoiu, V., Locovei, I. *Aplicaţii inginereşti ale calculatoarelor — optimizări.* Bucureşti, Editura didactică şi pedagogică, 1976.
9. Dumitrescu, St. *Automatizarea instalaţiilor petroliere din rafinării — partea întîi.* Bucureşti, Editura didactică şi pedagogică, 1967.
10. Dumitrescu, St. *Automatizarea instalaţiilor petroliere din rafinării.* Bucureşti, Editura didactică şi pedagogică, 1973.
11. Dumitrescu, St., Dumitrescu, I., Paiu, O. *Aplicaţii inginereşti ale calculatoarelor cu exemple din industria petrolieră şi petrochimică.* Bucureşti, Editura didactică şi pedagogică, 1974.
12. Georgescu, I. *Sisteme de echipamente şi programe.* Bucureşti, Editura tehnică, 1971.
13. Georgescu, I. *Sisteme de operare pentru calculatoarele numerice.* Bucureşti, Editura tehnică, 1974.
14. Gruenberger, F. *Computers and Communications-Toward a Computer Utility.* Englewood Cliffs, Pretince-Hall, 1968.
15. Ionescu, Al., Lupaş, L. *Tehnici de calcul în teoria sistemelor.* Bucureşti, Editura tehnică, 1974.
16. Kochenburger, R. L. *Computer Simulation of Dynamic Systems.* New-York, Prentice-Hall, 1972.
17. Kollence, K. W. *A Software View of Measurement Tools.* Datamation, vol. 17, nr. 1/ January, 1971.
18. Korn, A. G. *Recent Computer-System Developments and Continuous System Simulation.* Prague, Proceedings of the 7-th AICA Conference, August 1973.

Scanned with OKEN Scanner

19. Martin, J. *Teleprocessing Network Organization.* Englewood Cliffs, Prentice-Hall, 1970.
20. Mirtes, B. *Trends in the Desing of Special-Porpose Computers for Continuous Signal Processing.* Prague, Proceedings of the 7-th AICA Conference, August 1973.
21. Niculescu-Mizil, E. *Cibernetica și aplicațiile ei în tehnică și în economia industriei.* București, Editura Academiei R.S.R., 1973.
22. Pescaru, V., Dumitrescu, I., Bilciu, C., Satran, I., Nica, A. *Inițiere în teleprelucrarea datelor.* București, Editura tehnică, 1972.
23. Petrescu, A. *Calculatoare automate și programare.* București, Editura didactică și pedagogică, 1973.
24. Savas, E. S. *Conducerea cu calculatoare a proceselor industriale.* București, Editura tehnică, 1969.
25. Schaefer, W. *The Use of Electronic Computer for the Control of Chemical Processes.* Tutzing First European Symposium of DECHEMA, 5—6 March, 1968.
26. Yourdon, E. *Design of On-line Computer Systems.* Enghewood Cliffs, Prentice-Hall, 1972.
27. *** *IBM* 1800 *Functional Characteristics.* IBM—SRL, Form A22—6868, 1966.
28. *** *An Introduction to Engineering Analysis for Computers.* IBM—GIM, Form F20-8077, 1968.
29. *** *IBM System/7. System Sumary.* IBM, Form GA34—0002—2, 1972.
30. *** *IBM System/7. Functional Characteristics.* IBM, Form GA34—0003, 1972.
31. *** *Manuel du Software.* Paris, CAP, 1973.
32. *** *Calculateur T2000 — Systèmes d'entrées-sorties.* Paris, Telemecanique, MSP—D6560, 1969.
33. *** *IBM Data Processing Application. Plant Monitoring and Control — manufacturing industry.* IBM, Form GE19—5011—0, 1971.
34. *** *PICS — Production Information and Control System.* IBM, Form E20—0280, 1971.
35. Bellman, R. E., Dreyfus, E. S. *Programarea dinamică aplicată.* București, Editura didactică și pedagogică, 1967.
36. Lee, T. H., Adams, G. E., Gaines, W. M. *Computer Process Control: Modeling And Optimization.* New York, London, Sydney, John Wiley & Sons.
37. Luyben, W. L. *Process Modeling, Simulation and Control for Chemical Engineers.* New York, McGraw-Hill Book.
38. Eykhoff Pieter. *Sistem identification, parameter and state estimation.* New York John Wiley & Sons, 1974.
39. Wilkes V. M. *Sisteme de calcul cu acces multiplu.* București, Editura tehnică, 1974.

Scanned with OKEN Scanner

Tiraj: 3600
Ex. S.P.: 80 legate
Coli de tipar: 16,5
Hîrtia: Scris tip I A
Format: 16/70×100
Bun de tipar: 1.XI.1977
Ediţia: 1977
Nr. plan: 5694

Întreprinderea Poligrafică Cluj
Str. Brassai Nr. 5—7
Municipiul Cluj-Napoca
Republica Socialistă România
Comanda Nr. 327/1977

Scanned with OKEN Scanner